UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Grado en Ingeniería de la Salud

Mención Bioinformática

Diseño de un flujo de trabajo para el análisis de datos procedentes de secuenciación masiva

Design of a workflow for the analysis of massive sequencing data

Realizado por

Adrián Segura Ortiz

Tutorizado por

Prof. José Manuel Jerez Aragonés

Co-tutorizado por

Dra. Martina Álvarez Pérez

Departamento

Lenguajes y Ciencias de la Computación

Málaga, Junio de 2021

Escuela Técnica Superior de Ingeniería Informática

Grado en Ingeniería de la Salud

Mención Bioinformática

Diseño de un flujo de trabajo para el análisis de datos procedentes de secuenciación masiva

Design of a workflow for the analysis of massive sequencing data

Realizado por
Adrián Segura Ortiz

Tutorizado por
Prof. José Manuel Jerez Aragonés

Co-tutorizado por
Dra. Martina Álvarez Pérez

Departamento
Lenguajes y Ciencias de la Computación

Universidad de Málaga

Málaga, Junio de 2021

Abstract

Nowadays, massive sequencing has been integrated in many clinical laboratories because it is the most powerful tool to identify molecular alterations in patient samples. With this, a clear need has arisen to design software capable of processing the immense amount of data produced by the different sequencing equipment.

The workflow described in this project has been designed to run on the Picasso supercomputer [1] for the analysis of data from the Laboratorio de Biología Molecular del Cáncer [2], so its implementation is adapted to the methodology used in that center, that is, targeted sequencing with panels of amplicons using Ion Torrent single-read technology.

The script, implemented mainly in Bash, covers the usual stages of read processing, alignment, variant calling, as well as the detection of copy number alterations and genomic rearrangements. After its execution, the user obtains several tables in XLSX format with information about the variants detected for each of the samples. This automates the processing of the raw data from the sequencer and provides the user with a useful data source for subsequent clinical tasks such as target drug assignment.

Keywords: Workflow, Massive sequencing, Molecular alterations, Clinical laboratory, Ion Torrent.

Resumen

Actualmente, la secuenciación masiva ha sido integrada en numerosos laboratorios clínicos a causa de ser la herramienta más potente para llevar a cabo la identificación de alteraciones moleculares sobre muestras de pacientes. Con ello, ha surgido la clara necesidad de diseñar softwares capaces de procesar la inmensa cantidad de datos producidos por los diferentes equipos de secuenciación.

El flujo de trabajo descrito en este proyecto se ha destinado a su ejecución en la supercomputadora Picasso [1] para el análisis de datos procedentes del Laboratorio de Biología Molecular del Cáncer [2], por lo que su implementación se adapta a la metodología realizada en dicho centro, esto es, secuenciación dirigida con paneles de amplicones mediante tecnología Ion Torrent de lectura única.

El script implementado principalmente en Bash, abarca las usuales etapas de procesado de lecturas, alineamiento, llamada e identificación de variantes, así como la detección de alteraciones en el número de copias y reordenamientos genéticos. Tras su ejecución, el usuario obtiene diversas tablas en formato XLSX con información acerca de las variantes detectadas para cada una de las muestras. Con ello, se consigue automatizar el procesamiento de los datos brutos del secuenciador y se proporciona al usuario una fuente de datos útil para posteriores tareas de ámbito clínico como la asignación de fármacos diana.

Palabras clave: Flujo de trabajo, Secuenciación masiva, Alteraciones moleculares, Laboratorio clínico, Ion Torrent.

Agradecimientos

En primer lugar, quiero agradecer a mi tutor José Manuel Jerez Aragonés y cotutora Martina Álvarez Pérez por haberme guiado en este proyecto a través de su conocimiento y experiencia.

También dar las gracias a Alicia Garrido Aranda y todo el personal del laboratorio por su agradable trato y apoyo. Y por último, a Héctor Mesa Jiménez, Manuel Gonzalo Claros Díaz y Rafael Larrosa Jiménez por ayudarme durante el diseño y codificación del script final de este trabajo.

Índice

1. Introducción **13**

1.1. Estado del arte . . . 14

1.2. Motivación . . . 16

1.3. Objetivos . . . 17

1.4. Tecnologías usadas . . . 17

2. Material y métodos **19**

2.1. Equipo de secuenciación . . . 20

2.2. Tipos de variaciones genéticas a detectar . . . 20

2.2.1. Variantes de un solo nucleótido (SNVs) . . . 20

2.2.2. Inserciones y deleciones (INDELs) . . . 21

2.2.3. Alteraciones en el número de copias (CNAs) . . . 22

2.2.4. Reordenamientos genéticos . . . 22

2.3. Flujo de trabajo . . . 22

2.3.1. Fase 1: Procesamiento de lecturas . . . 24

2.3.2. Fase 2 (DNA): Mapeo con genoma humano de referencia . . . 24

2.3.3. Fase 3 (DNA): Procesamiento de las alineaciones . . . 25

2.3.4. Fase 4 (DNA): Detección de SNVs e INDELs . . . 26

2.3.5. Fase 5 (DNA): Anotación de variantes . . . 29

2.3.6. Fase 6 (DNA): Detección de CNAs . . . 31

2.3.7. Fase 7 (RNA): Detección de reordenamientos genéticos . . . 32

2.4. Herramientas integradas . . . 33

2.4.1. Alineadores . . . 33

2.4.2. Llamadores de variantes . . . 36

2.4.3. Anotadores . . . 39

2.4.4. Detectores de CNAs . . . 41

2.4.5. Detectores de reordenamientos genéticos . . . 42

2.5. Scripts para la unión de resultados . . . 44

2.5.1. SNVs e INDELs: process_annotation.R: . . . 44
2.5.2. CNAs: merge_cnv_outputs.R: . . . 46
2.5.3. Reordenamientos genéticos: merge_gene_fusion_outputs.R: . . . 47
2.6. Modificaciones debidas a Picasso . . . 48
2.7. Paralelización y multiprocesado . . . 49

3. Resultados y Discusión **53**
3.1. Distribución de ficheros . . . 53
3.2. Archivos intermedios generados . . . 55
3.2.1. Informes de calidad . . . 56
3.2.2. Diagramas de Venn . . . 59
3.3. Tablas finales . . . 63
3.3.1. SNVs + INDELs . . . 63
3.3.2. CNAs . . . 70
3.3.3. Reordenamientos genéticos . . . 73
3.4. Comparación con los resultados de Ion Reporter . . . 75
3.4.1. SNVs + INDELs . . . 75
3.4.2. CNAs . . . 78
3.4.3. Reordenamientos Genéticos . . . 80
3.5. Tiempo de ejecución . . . 82
3.5.1. Mapeadores . . . 82
3.5.2. Tiempo de ejecución por paciente . . . 84

4. Conclusiones y Líneas Futuras **89**
4.1. Conclusiones . . . 89
4.2. Líneas Futuras . . . 90
4.2.1. Paralelismo . . . 90
4.2.2. Complementariedad entre mapeadores y llamadores de variantes . . . 91
4.2.3. Aprendizaje computacional . . . 91
4.2.4. Mayor flexibilidad . . . 92
4.2.5. Interfaz gráfica . . . 92
4.2.6. Script de actualización . . . 93

Referencias bibliográficas **95**

Apéndice A. Manual de uso **111**

A.1. Modificación del archivo de configuración 111

A.2. Posicionamiento y renombrado de los archivos FASTQ 115

A.3. Ejecución del script 115

Índice de figuras

1. Tipos de variaciones genéticas 21
2. Diagrama flujo de trabajo 23
3. Estrategia para el consensuado de variantes 27
4. Bases de datos consultadas por cada anotador 40
5. Instalación de herramientas 49
6. Distribución de directorios de salida para cada paciente 54
7. Informes de calidad FastQC y MultiQC 57
8. Informes de calidad Qualimap individual 58
9. Informes de calidad Qualimap comparativo 59
10. Diagramas de Venn para la intersección de cada variant caller 61
11. Diagrama de Venn para la unión de intersecciones 62
12. Tabla final SNVs e INDELs parte 1 64
13. Tabla final SNVs e INDELs parte 2 66
14. Tabla final SNVs e INDELs parte 3 67
15. Tabla final SNVs e INDELs parte 4 69
16. Tabla final CNAs ejemplo de deleción y duplicación 71
17. Gráficas generadas por CnvKit 72
18. Tabla final de Fusiones Génicas 73
19. Informe generado por Arriba 74
20. Comparación de SNVs e INDELs con Ion Reporter 76
21. Comparación de CNAs con Ion Reporter 79
22. Comparación de Fusiones Génicas con Ion Reporter 81
23. Intersección de los cuatro mapeadores finalistas 84

Índice de cuadros

1. Enlaces de identificadores anotados 70
2. Comparación de SNVs e INDELs con Ion Reporter 77
3. Comparación de Fusiones Génicas con Ion Reporter 82
4. Tiempo de ejecución para cada alineador 83
5. Tiempo de ejecución para un paciente 86

1

Introducción

La **secuenciación masiva** (*Next Generation Sequencing, NGS*) constituye actualmente el análisis más detallado y eficaz en la tarea de identificar variantes producidas en el material genético. Estas mutaciones han demostrado estar habitualmente asociadas a diversas enfermedades y patologías, siendo este el principal motivo por el que numerosos centros han introducido esta tecnología en su práctica clínica diaria con el fin de individualizar el diagnóstico, pronóstico y tratamiento de los pacientes [3, 4, 5].

Las diversas plataformas de secuenciación masiva disponibles hoy en día se caracterizan por emplear distintos principios químicos y técnicas que generan ciertas diferencias cuantitativas y cualitativas en sus resultados. Los dos tipos de secuenciación más conocidos y empleados en la actualidad son **Illumina** [6] e **Ion Torrent** [7], considerados los dos grandes pilares de la secuenciación moderna.

Además, existen diversas estrategias respecto a las zonas que se pretenden secuenciar dependiendo de la finalidad del estudio. Estas son, secuenciación del genoma completo (*whole genome sequencing, WGS*), secuenciación de la totalidad del exoma (*whole exome sequencing, WES*) y secuenciación dirigida mediante paneles genéticos.

La aparición de estas técnicas que permiten leer en paralelo millones de fragmentos de DNA, han revolucionado sin duda la microbiología, la cual ha dejado de tener un ámbito exclusivamente laboratorial y ha incorporado un imprescindible componente computacional. La clara necesidad de tener que procesar abundantes cantidades de datos producidos por los secuenciadores masivos, junto con la cada vez más compleja tarea de análisis que se debe realizar con los mismos, ha hecho imprescindible el papel de los **bioinformáticos** en los laboratorios

clínicos.

El proceso bioinformático habitual en este tipo de estudios abarca las etapas asociadas al procesamiento de las lecturas, alineamiento con el genoma humano de referencia, llamada de variantes y anotación. Además, las principales alteraciones genéticas que se suelen detectar son variaciones de un único nucleótido (*single nucleotide variant, SNV*), pequeñas inserciones y deleciones (*INDEL*), alteraciones en el número de copias (*copy number alteration, CNA*) y reordenamientos genéticos. Para automatizar este proceso, es necesario el uso de numerosas herramientas computacionales y estadísticas ejecutadas bajo una secuencialidad predefinida, es decir, la puesta en marcha de un **flujo de trabajo**. Los laboratorios clínicos moleculares que llevan a cabo estudios con datos procedentes de secuenciación NGS, tienen la opción de utilizar uno o varios flujos de trabajo, ya estén construidos a medida por el propio laboratorio o proporcionados por la plataforma de secuenciación.

1.1. Estado del arte

A continuación, se exponen algunos ejemplos de flujos de trabajo y aplicaciones software capaces de llevar a cabo el procesamiento automático de datos NGS.

Respecto a los softwares ofrecidos por las propias casas comerciales, cabe destacar **Ion Reporter** [8, 9] ya que se trata del programa más empleado para el tipo de secuenciación de Ion Torrent. Tal y como expone su documentación, cubre el análisis completo de los datos, incluyendo de forma adicional varias interfaces gráficas que facilitan visualizar tanto el progreso como el resultado de la ejecución. Además, permite llevar a cabo la generación de varios informes y la exportación de los mismos para poder ayudar en posteriores tareas por parte del laboratorio. Respecto a su implementación, el software afirma priorizar el proceso de llamada de variantes y su posterior anotación haciendo uso de un total de más de 25 bases de datos distintas, a lo cual se le añade la posibilidad de incluir algunos plugins adicionales que se pueden configurar para que formen parte del flujo de trabajo a ejecutar.

No obstante, al igual que suele ocurrir con el resto de programas asociados a otros se-

cuenciadores como Illumina (**DRAGEN**) [10, 11] o roche 454 (**NextGENe**) [12, 13], el flujo de trabajo se encuentra preconfigurado independientemente de las funcionalidades adicionales que posteriormente puedan encontrarse disponibles. Es por esto, que muchos laboratorios deben adaptarse a las especificaciones del software, a pesar de que debería ser el flujo de trabajo el que se amoldara a la actividad realizada en el centro. Por otro lado, la mayoría de estos programas se encuentran condenados al uso de determinadas herramientas como mapeadores o llamadores de variantes que no siempre resultan ser los más aconsejables y que además en muchas ocasiones han sido implementadas por la propia casa comercial.

En el lado opuesto, se encuentran una amplia variedad de scripts implementados por bioinformáticos o entidades privadas con el fin de proporcionar una alternativa viable ante los inconvenientes expuestos anteriormente, siendo alguno de ellos de código abierto y otros de adquisición no gratuita. Además, cada pipeline tiene su propio nivel de especificación ante las distintas modalidades de secuenciación, sigue su propia estrategia a la hora de usar varias herramientas y abarca distintas fases del flujo completo.

En primer lugar, está el ejemplo de **BRB-SeqTools** [14], el cual se define como un pipeline tool de fácil manipulación que hace uso de varias aplicaciones software conocidas con el fin de proporcionar ayuda a científicos en el análisis y procesamiento de datos procedentes de secuenciación NGS. Admite la importación de datos tanto de RNA-Seq como de DNA-Seq, ofreciendo la posibilidad de llevar a cabo su alineamiento mediante el uso de varios mapeadores. Respecto a la llamada de variantes, se incluyen diversas herramientas con el objetivo de ofrecer un resultado fiable que facilite la posterior anotación, para lo cual se emplean anotadores de mutación somática incluidas en el propio flujo.

Con esto, a diferencia de muchos softwares de casas comerciales, permite al personal de laboratorio tener la posibilidad de elegir diferentes herramientas a lo largo del flujo mediante una sencilla interfaz gráfica destinada a realizar la configuración del sistema. No obstante, el flujo en ningún caso puede acomodarse a la modalidad de secuenciación del laboratorio ya que proporciona un servicio genérico que no permite optimizar los parámetros de las herramientas a las necesidades de diferentes tipos de estudio. Por otro lado, el uso de este pipeline tool, provoca que el laboratorio dependa por completo del equipo de mantenimiento de la aplicación

para poder acceder a nuevas herramientas o actualizaciones, las cuales se llevarán a cabo de manera independiente a las necesidades del laboratorio donde se emplee.

En segundo lugar, se debe mencionar a la máquina virtual **TREVA** [15]. Esta herramienta, en relación a este trabajo, se especializa en la detección de variantes genómicas asociadas a estudios de cáncer. Respecto a su implementación, cubre las mismas fases del trabajo que el pipeline anterior pero sin contemplar el análisis de lecturas procedentes de RNA-Seq. Además, la mayoría de las herramientas contenidas en la máquina virtual coinciden con las interiorizadas en BRB-SeqTools, sin embargo, en este caso no se permite un proceso de activación y desactivación tan cómodo como en la ocasión anterior.

Por tanto, por las desventajas mencionadas hasta el momento, se explica el hecho de que muchos laboratorios opten por la construcción de un flujo de trabajo diseñado a medida con el fin de optimizar sus resultados. No obstante, es importante recalcar el uso habitual de varias opciones con el objetivo de poder comparar los resultados obtenidos por cada pipeline disponible, lo cual suele mejorar la fiabilidad de los datos y aumentar la confianza de los especialistas a la hora de elaborar informes clínicos.

1.2. Motivación

En primer lugar, la principal causa que promueve la elaboración de este trabajo es la mejora de la capacidad y eficacia por parte del laboratorio respecto al procedimiento asociado al análisis de los datos de secuenciación y las posteriores tareas de ámbito clínico como la asignación de fármacos diana correspondientes a cada perfil. Esto, además de beneficiar el funcionamiento del laboratorio afecta por supuesto a los cientos de pacientes cuyas muestras son analizadas semanalmente. Realizar un flujo de trabajo cuyos resultados puedan tener la capacidad de beneficiar la salud de otras personas, es sin duda la motivación más importante que puede tener un proyecto bioinformático.

A nivel profesional y académico, el interés que despierta el aprendizaje asociado a la elaboración de este trabajo es indiscutible, tanto en el ámbito computacional como en la con-

templación de un ambiente laboral propio de un laboratorio real. Aprender el funcionamiento de numerosas herramientas asociadas a las técnicas biológicas vistas a lo largo de la titulación, permite llevar a la práctica muchos de los conocimientos adquiridos hasta el momento y enfrentarse a problemas reales que requieren una solución efectiva.

Por otro lado, la ferviente aparición de la medicina de precisión en el ámbito clínico representa otra clara motivación de este proyecto. El aumento de demanda propia de esta metodología ha hecho necesaria la elaboración de numerosos flujos de trabajo que traten de automatizar la detección de variantes mediante la concatenación de múltiples herramientas especializadas en este tema.

1.3. Objetivos

Diseñar un script personalizado y flexible que permita el análisis de datos procedentes de secuenciación masiva dirigida con el panel de 161 genes (tecnología de amplicones de Ion Torrent) para la identificación de alteraciones moleculares en pacientes oncológicos.

Comparar las variantes identificadas por el script personalizado con las encontradas por el software comercial (Ion Reporter). Esto permitiría a los profesionales encargados del análisis de los datos de secuenciación, una doble lectura de los resultados obtenidos de manera automatizada.

1.4. Tecnologías usadas

El flujo de trabajo se encuentra diseñado principalmente en Bash para su rutinaria ejecución en la **supercomputadora Picasso** [1], no obstante, incluye herramientas implementadas en múltiples lenguajes de programación como R, Python o Perl. Algunas de ellas se caracterizan por su instalación y uso directo mientras que otras requieren de su codificación manual, tal y como pasa con muchas de las herramientas disponibles a través de paquetes de R.

Además, algunos de los pasos que interconectan las salidas con las entradas de varias de

estas herramientas, han sido abordados mediante la elaboración de scripts propios en R, siendo muy habitual su uso para la unión consensuada de resultados y la resolución de algunos problemas de compatibilidad.

Por otro lado, en ocasiones puntuales ha sido necesaria la modificación de algunos scripts complementarios a las herramientas, tanto de Python como de Bash, con el fin de hacer viable su ejecución en Picasso. Por motivos de restricción, algunas aplicaciones tuvieron que ser instaladas localmente y por tanto mostradas al resto mediante la notificación de su nueva ruta de acceso.

Por último, cabe mencionar el uso y aprendizaje de LaTeX para la redacción de esta memoria, junto con la aplicación web creately [16] que ha permitido el diseño de múltiples diagramas, entre los que se encuentra el correspondiente al flujo de trabajo.

2

Material y métodos

El script implementado trata de **adaptar y concatenar las herramientas gratuitas disponibles** a las necesidades específicas de la modalidad de secuenciación realizada en el laboratorio, para lo cual se requiere llevar a cabo una optimización de sus correspondientes parámetros a la vez que se cuida el usual equilibrio entre la calidad de los resultados y el tiempo de ejecución necesario.

Por otro lado, dentro de la técnica de secuenciación pertinente, se pretende otorgar al programa la mayor flexibilidad posible, de tal forma que el personal del laboratorio tenga cierto dominio en la ejecución del código. Para ello, se le permite modificar un sencillo **fichero de configuración** con el objetivo de escoger entre la aplicación de varias herramientas de igual fin o incluso solicitar la intersección de varias de ellas para lograr resultados de mayor fiabilidad. Además, en caso de que lo deseen, se permite la cancelación de ciertas fases dispensables en el flujo como la generación de informes de calidad, la representación de diagramas de Venn o la detección de algún tipo de alteración genética concreta (CNAs o fusiones génicas).

A modo de ayuda, se le proporciona la información necesaria acerca de dichas herramientas, un fichero de configuración predeterminado y contará tras cada ejecución con un documento informativo acerca del tiempo de ejecución que ha requerido cada fase del flujo para cada muestra.

De esta forma, el laboratorio será capaz de poner a punto el pipeline construido gracias a la experiencia que adquieran con su uso a lo largo del tiempo o incluso elaborar distintos ficheros de configuración en función del tipo de muestra, el tiempo o la fiabilidad de los resultados que deseen obtener.

2.1. Equipo de secuenciación

El Laboratorio de Biología Molecular del Cáncer apuesta por la tecnología **Ion Torrent** de lectura única basada en el uso de un secuenciador semiconductor que detecta los cambios de pH producidos tras la incorporación de cada nucleótido. Una de las principales consecuencias de esta elección a nivel bioinformático, es la generación de un único fichero FASTQ por muestra, lo cual condiciona el uso de numerosas herramientas contenidas en el flujo de trabajo construido.

Los datos proporcionados por el laboratorio proceden de una secuenciación que emplea el panel Oncomine Comprehensive Assay v3 (Ion Torrent by Thermo Fisher Scientific), el cual cubre un total de 161 genes relevantes de cáncer. Se trata de una secuenciación dirigida con amplicones, que a diferencia de las capturas de híbridos, producen numerosas lecturas de cada región que pueden ser malinterpretadas como duplicados de PCR por algunas herramientas.

2.2. Tipos de variaciones genéticas a detectar

Tras la preparación de las muestras y la correspondiente secuenciación de las mismas, los principales tipos de variaciones que tratan de detectar los estudios del laboratorio son los siguientes: Variantes de un solo nucleótido (*Single Nucleotide Variants* o SNVs), Inserciones y deleciones (*Insertions and Deletions* o INDELs), Alteraciones en el número de copias (*Copy Number Alterations* o CNAs) y Reordenamientos genéticos.

2.2.1. Variantes de un solo nucleótido (SNVs)

Los SNVs son variaciones en la secuencia de DNA que provocan el cambio de un nucleótido por otro. El efecto de estos cambios va a variar en función de su localización y características. El principal motivo que suele determinar si un cambio de nucleótido resulta ser trascendental o no en el organismo, depende de si acarrea una sustitución de aminoácido durante el proceso de traducción. En dicho caso, la proteína resultante puede verse alterada mediante el cambio

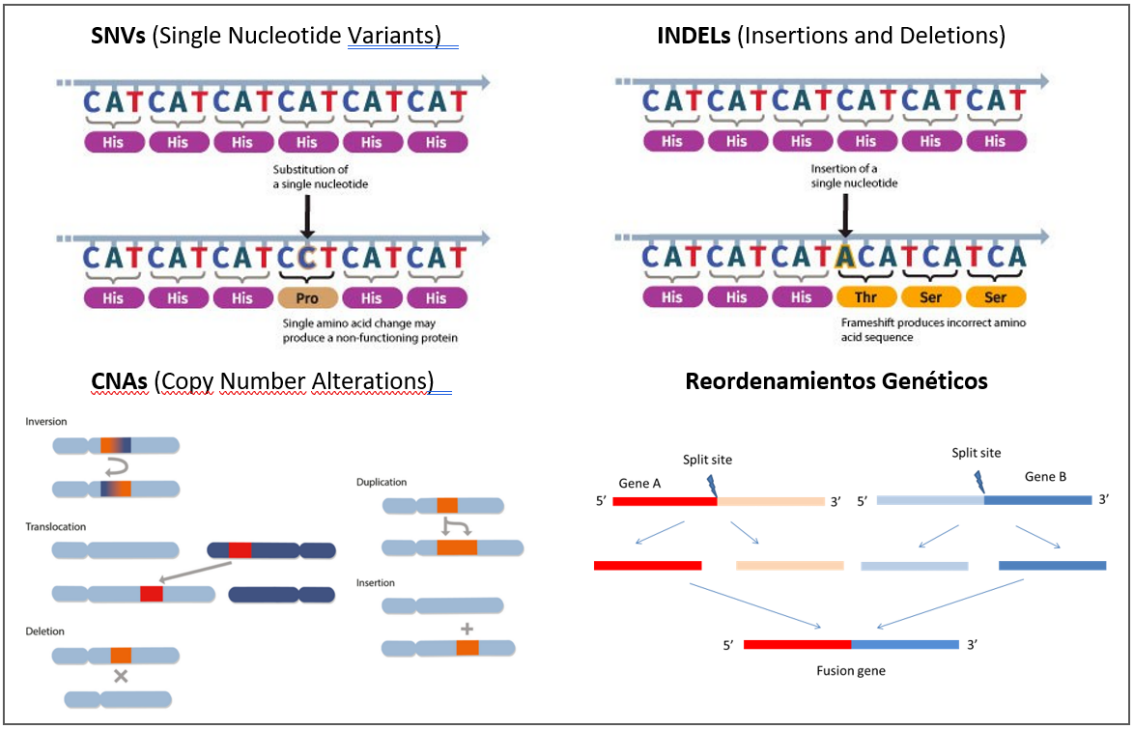


Figura 1: El flujo de trabajo se encuentra destinado a la identificación de cuatro tipos de alteraciones genéticas distintas: SNVs, INDELs, CNAs y Reordenamientos Genéticos. Imágenes tomadas de [17, 18]

de su estructura, su funcionalidad o su nivel de expresión.

Cuando este tipo de variante posee una frecuencia superior al 1 % en la población, pasa a denominarse polimorfismo o SNP (Single Nucleotide Polymorphism), lo cual no resulta interesante en el ámbito clínico y por tanto suele ser descartado del posterior estudio mediante procesos de filtrado.

2.2.2. Inserciones y deleciones (INDELs)

Los INDELs hacen referencia a pequeños segmentos de DNA que han sido insertados o eliminados en el genoma. Su longitud es normalmente inferior a 50 pb, sin embargo, es fundamental conocer su tamaño exacto para poder predecir su repercusión en el paciente. Cualquier INDEL con una longitud que no sea múltiplo de tres, significará el desajuste del marco de lectura para todos los codones posteriores, dando lugar a una incorrecta traducción del resto de bases que puede suponer grandes cambios en la proteína producida.

2.2.3. Alteraciones en el número de copias (CNAs)

Los CNAs representan un conjunto de variaciones estructurales que tienen como consecuencia un cambio en el número de copias de un gen o segmento de DNA particular dentro del genoma. Aunque todavía no se conoce con precisión lo que estas variaciones contribuyen a la enfermedad humana, sí que han sido experimentalmente asociadas a diversos cánceres cuando se detecta un elevado número de copias en algunos genes concretos.

2.2.4. Reordenamientos genéticos

Los reordenamientos genéticos tienen como resultado la aparición de nuevos genes producidos por la yuxtaposición de secuencias de DNA procedentes de igual gen (fusión intragénica) o de dos genes separados (fusión intergénica). Este fenómeno suele ocurrir durante reordenaciones cromosómicas en las que parte del DNA de un cromosoma se transfiere a otro cromosoma.

Al igual que en el caso anterior, estas variaciones se encuentran habitualmente asociadas al ámbito de la oncología clínica, no obstante, a diferencia del resto de variaciones comentadas anteriormente, la detección de fusiones génicas se realiza en este caso mediante el estudio de la muestra de RNA de los pacientes.

2.3. Flujo de trabajo

El pipeline diseñado consta principalmente de 7 etapas que abarcan desde el análisis de las lecturas brutas del secuenciador hasta la exportación de las variantes identificadas en formato XLSX.

El programa comienza su funcionamiento a través de la detección de archivos FASTQ en el directorio principal de trabajo, de tal forma que cada fichero contenga el **ID de su paciente** y el **tipo de muestra** (dna o rna) separados por un **guion bajo**. Por ejemplo: p1_dna.fastq.

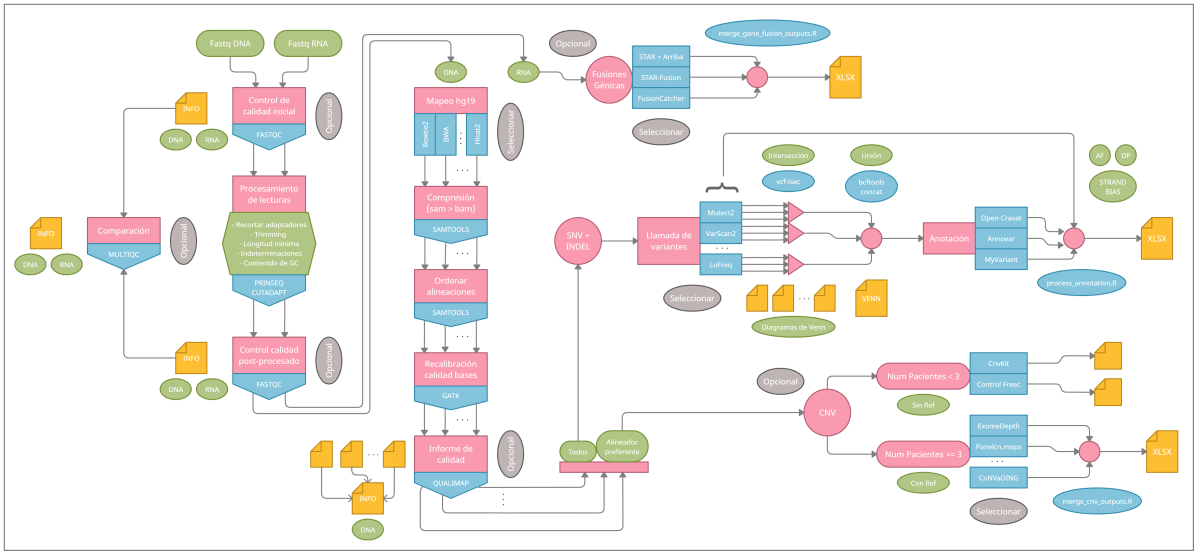


Figura 2: Se presenta el diagrama completo del flujo de trabajo elaborado en este proyecto. En color rosa se representan cada uno de los pasos importantes a ejecutar en el flujo, de color azul las herramientas empleadas en cada tarea, de color verde detalles destacables acerca de cada fase, de color gris indicadores de selección u opcionalidad y por último de color naranja los archivos tanto intermedios como finales de salida. La primera columna hace referencia al procesamiento inicial de lecturas, la segunda muestran las tareas de mapeo y procesamiento de su resultado y cada una de las líneas de la derecha representan el procedimiento a seguir para la detección de cada tipo de alteración genética.

Además, deben estar correctamente emparejados ya que cada paciente debe tener su respectiva muestra tumoral de DNA y de RNA, es decir, para p1_dna.fastq debe existir un p1_rna.fastq. Tras esto, el script se encarga de crear las carpetas correspondientes a cada paciente detectado y desplazar los archivos FASTQ a su nuevo destino.

Una vez construido el punto de partida, el flujo de trabajo avanza completando cada etapa para todas las muestras. Esto significa que el script nunca ejecutará la siguiente fase sin que todos los pacientes hayan completado previamente la anterior.

El motivo de realizar esta estrategia consiste en la necesidad de poseer todas las lecturas mapeadas para poder construir una referencia en la detección de CNAs. Dicha fase es un punto especial del script donde los caminos de todas las muestras se cruzan y se requiere que todas ellas se encuentren en el mismo punto de la ejecución. A continuación, se exponen los detalles de cada una de las fases que constituyen el flujo de trabajo.

2.3.1. Fase 1: Procesamiento de lecturas

En esta primera fase del flujo de trabajo, las lecturas brutas del secuenciador son analizadas, modificadas y filtradas con el fin de eliminar posibles artefactos artificiales procedentes del equipo de secuenciación que puedan llegar a ocasionar falsos positivos durante la detección de variantes.

Para ello, la herramienta **Cutadapt** [19, 20] es utilizada en primer lugar para la eliminación de adaptadores, cuyas secuencias han sido facilitadas por el propio equipo técnico de **ThermoFisher** [21]. El resultado de esta operación es dirigido a **PRINSEQ** [22, 23], la cual se encarga de analizar las calidades de las lecturas y proceder a la eliminación de indeterminaciones, recorte de extremos de baja calidad (trimming) y filtrado de las restantes en función de su longitud y contenido GC.

Con el fin de que el usuario pueda visualizar la mejora obtenida, el flujo integra el uso opcional de las herramientas **FastQC** [24] y **MultiQC** [25, 26], las cuales permiten la generación de informes con múltiples gráficas que tratan de comparar las calidades de las lecturas antes y después del procesado.

2.3.2. Fase 2 (DNA): Mapeo con genoma humano de referencia

En esta etapa, las lecturas procesadas de las muestras de DNA son alineadas sobre el genoma humano de referencia con el fin de poder ubicarlas y contrastarlas con la información conocida para facilitar la posterior identificación de alteraciones moleculares en las muestras tumorales de los pacientes.

Dado que el laboratorio desea comprobar los resultados con los obtenidos por Ion Reporter, la referencia elegida para realizar esta tarea debe ser la misma que la empleada en su equipo, esto es, **hg19** [27]. El uso de cualquier otra referencia más actualizada como hg38, además de dificultar la comparación entre resultados, sería incompatible con las coordenadas de los exones definidos en sus paneles de secuenciación, ya que en dicho caso no harían referencia

a las regiones realmente secuenciadas y confundiría a todo el proceso bioinformático.

El flujo integra un total de 8 mapeadores distintos, de los cuales el usuario puede elegir uno o varios para ser ejecutados sobre sus muestras. En caso de elegir múltiples alineadores, a cambio de aumentar en cierta medida el tiempo de ejecución, el programa utiliza la información proporcionada por todos ellos con el fin de dar en fases posteriores un resultado consenso de mayor fiabilidad.

Antes del alineamiento, todos los mapeadores necesitan generar su propio índice del genoma de referencia, que en este caso al tratarse de un FASTA invariable (hg19.fa), sólo será construido en la primera puesta en marcha del script. De esta forma, una vez producido el índice de un mapeador concreto, el flujo de trabajo reconocerá su existencia en las posteriores ejecuciones para evitar su sobrescritura y pérdida de tiempo de ejecución.

Los mapeadores implementados en el flujo han sido seleccionados tras una recopilación bibliográfica de artículos que tratan de comparar su eficiencia y calidad de resultados [28, 29, 30, 31]. La lista definitiva de herramientas contiene a **BWA** [32, 33], **BOWTIE2** [34, 35, 36], **GMAP** [37], **Subread** [38, 39], **HISAT2** [40, 41], **NovoAlign** [42], **GEM3** [43] y **Kart** [44], de tal forma que el usuario pueda seleccionar el mapeador o la combinación de los mismos que más se adecúe a sus muestras.

2.3.3. Fase 3 (DNA): Procesamiento de las alineaciones

Tras el mapeo de las lecturas, los archivos resultantes de los alineadores deben ser preparados para la posterior detección de variantes. En este proceso se lleva a cabo la compresión de los ficheros SAM, la ordenación de las lecturas mapeadas en función de sus coordenadas y por último la recalibración de la calidad de las bases. Para ello, en las dos primeras operaciones se emplea **Samtools** [45] y por último **GATK BaseRecalibrator** [46].

La **recalibración de la calidad de las bases** es un procedimiento que hace uso del aprendizaje automático con el objetivo de ajustar los puntajes establecidos por la máquina de secuenciación. En ocasiones, los equipos de laboratorio tienden a subestimar o sobreestimar la

calidad de ciertas bases, las cuales pueden provocar la aparición de falsas alteraciones genéticas en los resultados.

En concreto, la metodología llevada a cabo por la herramienta de GATK [47] se nutre de diversas métricas, algoritmos y estrategias a la vez que analiza información acerca de variantes conocidas disponibles en bases de datos como **dbSnp** [48, 49] y **GnomAD** [50, 51]. Como resultado, GATK consigue otorgar una mayor precisión a los puntajes de calidad y por tanto aumentar la fiabilidad de posteriores tareas en el flujo de trabajo.

Al igual que en la primera fase, el usuario tiene la posibilidad de generar informes de calidad sobre los resultados obtenidos. En este caso, el pipeline integra el uso opcional de **Qualimap** [52], la cual se encarga del análisis de los archivos BAM generados con el fin de medir la calidad de sus alineaciones y posteriormente realizar un informe comparativo entre los resultados de los distintos mapeadores seleccionados.

2.3.4. Fase 4 (DNA): Detección de SNVs e INDELs

Una vez procesadas las lecturas alineadas, se da paso a las herramientas encargadas de detectar SNVs e INDELs, es decir, los llamados **Variant Callers**. Antes de exponer la selección realizada, es necesario abordar una distinción previa en este tipo de herramientas. Por un lado, están aquellas que necesitan las lecturas mapeadas del tejido sano del paciente, mientras que otras prescinden de ella a cambio de obtener una menor precisión en los resultados. Lamentablemente, por motivos económicos el Laboratorio de Biología Molecular del Cáncer no secuencia el tejido sano de los pacientes en la tarea de llevar a cabo la detección de alteraciones genéticas. Es por esto, que la elección solo puede realizarse entre las herramientas que permiten no hacer uso de dicha información.

Al igual que con el caso de los mapeadores, el flujo integra el uso de múltiples opciones que han sido contempladas gracias a la recopilación previa de la bibliográfica pertinente [53, 54, 55, 56, 57], de cuya lectura se han seleccionado un total de 6 variant callers distintos. Estos son **GATK Mutect2** [58], **GATK Haplotypecaller** [59], **VarScan2** [60, 61, 62], **VarDict** [63, 64], **Freebayes** [65, 66, 67] y **LoFreq** [68].

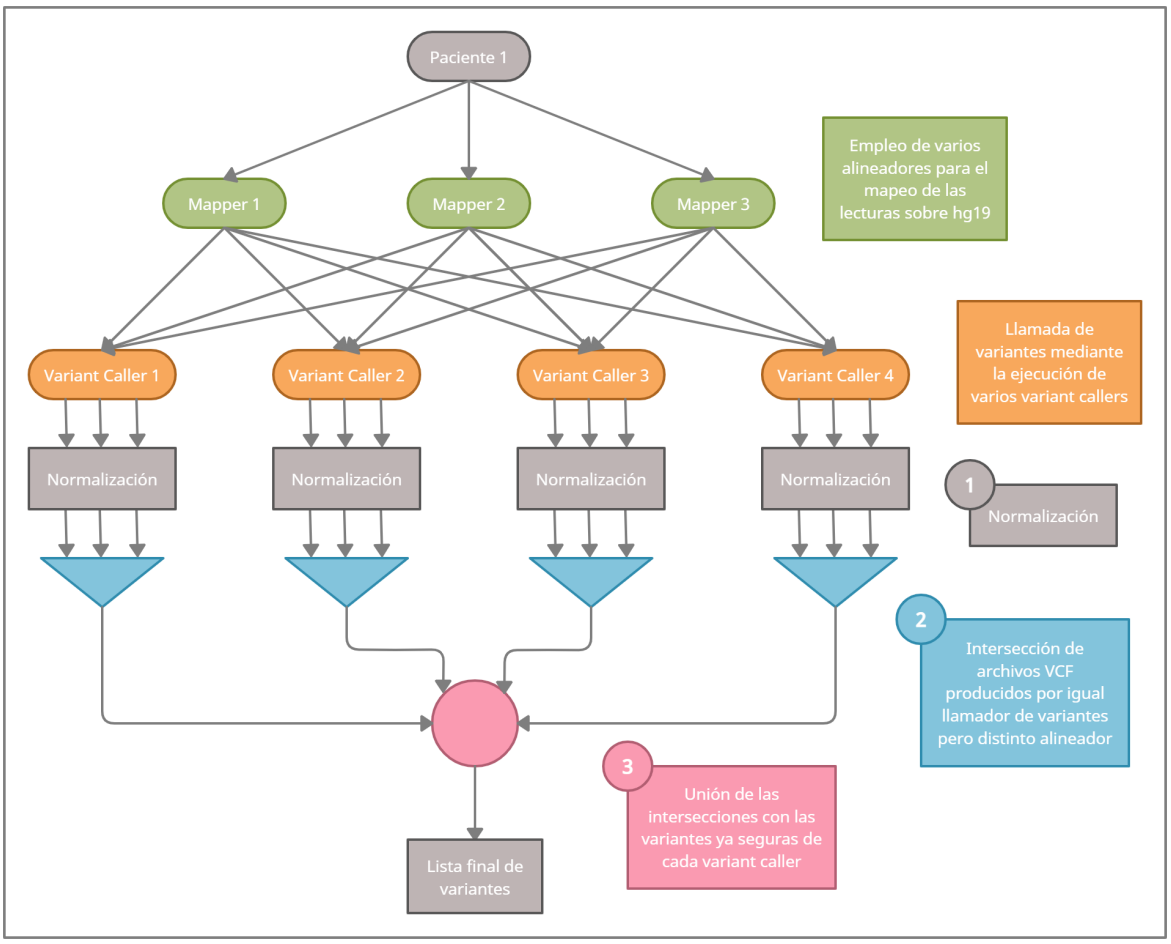


Figura 3: Tras la generación de todos los archivos VCF (verde y naranja), se lleva a cabo una estrategia de consensuado dividida en tres pasos: 1. Normalización (gris) 2. Intersección de variantes de igual llamador pero distinto mapeador (azul) 3. Unión de intersecciones (rosa).

La estrategia diseñada para consensuar las variantes detectadas por todos los variant callers y a su vez procedentes de múltiples mapeadores, se caracteriza por un procedimiento dividido en tres fases expuesto en la figura 3. En primer lugar, los archivos VCFs resultantes de la ejecución de todas las combinaciones son normalizados con el fin de que solo exista una única representación posible para cada variante. Con ello, se consiguen optimizar las columnas de referencia y alteración a la vez que se restringe el uso de una única línea por variante aunque compartan la misma posición en el genoma.

En segundo lugar, el script lleva a cabo una intersección de las variantes identificadas por igual llamador de variantes pero distinto alineador. Con el fin de que un mal mapeador no

pueda afectar en gran medida a la precisión de los resultados, para los casos en el que el número de alineadores ejecutados sea mayor que dos, se permite la abstención de las variantes en un fichero VCF, por lo que si todos los mapeadores a excepción de uno detectan una alteración, esta será incluida en los resultados. Como es lógico, en caso de ejecutar un único mapeador la intersección será su propia salida sin ninguna modificación. El objetivo de esta tarea reside en conseguir la eliminación de futuros falsos positivos causados por un mal alineamiento y aumentar la fiabilidad de los resultados.

Tras ello, se realiza una unión de dichas intersecciones que contienen las variantes ya seguras de cada variant caller. Su finalidad se centra en aprovechar la diversidad de los algoritmos de búsqueda y obtener un mayor número de alteraciones genéticas.

Esta estrategia también se recoge de forma similar en la publicación de Maurizio Callari et al. [69] donde se realizan varias intersecciones y combinaciones de variantes mediante el uso de múltiples herramientas. Además, debido a la poca concordancia entre diversos variant callers, en algunos artículos que tratan de comparar sus resultados, como el de el de Ze Kun Liu et al. [56], acaban exponiendo el uso conjunto de todos ellos como la opción más adecuada.

Para ejecutar estas tres tareas de la forma más eficiente posible se hace uso de las herramientas **bcftools norm** [70] para la normalización, **vcftools vcf-isec** [71] para las intersecciones y **bcftools concat** [72] para el proceso de unión.

Sin embargo, estas dos últimas herramientas poseen el inconveniente de ignorar cierta información acerca de los valores de **frecuencia alélica**, **profundidad de lectura** y **strand bias** de las variantes contenidas en los distintos VCFs. Por un lado, vcf-isec decide mantener para cada variante únicamente los valores que lea en el primer archivo que la contenga, ignorando los valores generados por el resto de mapeadores. Por otro lado, bcftools concat al unir VCFs procedentes de distintos variant callers que poseen ciertas diferencias en sus estructuras, en ocasiones se generan conflictos con estos valores que la herramienta termina solventando mediante la omisión de sus contenidos.

Dado que estas desventajas en su uso no afectan en ningún momento a la identificación de las variantes, no suponen ningún riesgo para la posterior anotación. En otras palabras, la

pérdida de información nunca repercute a las columnas referentes al cromosoma, posición, referencia y alteración, las cuales son las únicas requeridas por los anotadores para llevar a cabo su trabajo. De igual forma, la información extraviada será posteriormente recuperada en fases posteriores durante la generación del XLSX final gracias a la implementación pertinente de un script de R.

El motivo por el que se realiza este procedimiento se debe a que el tiempo que requeriría realizar las intersecciones y uniones manualmente, supera con diferencia el tiempo necesario para la recuperación de los valores anteriormente descritos.

Con el fin de que el usuario pueda contemplar visualmente el resultado de este procedimiento, el flujo de trabajo integra el uso opcional de **vcftoolz compare** [73], el cual se encarga de graficar distintos diagramas de Venn que permiten estudiar el solapamiento entre las distintas salidas de las herramientas. La generación de estas gráficas le puede resultar bastante útil al usuario, ya que le proporcionan información que le ayuda a poner a punto el script en posteriores ejecuciones. Entre otras observaciones, el usuario puede identificar mapeadores discrepantes del resto o variant callers innecesarios a causa de no poseer un campo de búsqueda propio.

2.3.5. Fase 5 (DNA): Anotación de variantes

Tras obtener la lista definitiva de variantes, el siguiente paso consiste en llevar a cabo la anotación de las mismas. Para ello, se utilizan tres herramientas inicialmente independientes de igual fin: **ANNOVAR** [74, 75], **Open Cravat** [76, 77] y **MyVariant** [78, 79, 80]. Su funcionamiento habitual se basa en la lectura del archivo VCF resultante de la fase anterior y consultar información de cada variante en múltiples bases de datos.

Para llevar a cabo la anotación tanto con ANNOVAR como con Open Cravat es necesario realizar la descarga previa de los archivos correspondientes a dichas bases de datos, los cuales deben ser revisados y actualizados con cierta frecuencia. Por el contrario, el funcionamiento de MyVariant se basa en peticiones tipo GET que requieren de acceso a internet durante la ejecución del programa. A pesar de que esto posea la ventaja de poder prescindir se su constante

mantenimiento, no es una metodología viable para el flujo de trabajo si se tiene en cuenta de que los nodos de Picasso en los que se ejecuta el script no poseen conexión a internet.

Para solucionar este problema se adaptó el uso habitual de MyVariant al procedimiento llevado a cabo por el resto de anotadores. La forma de alcanzar este objetivo, consiste en almacenar en un RData el resultado de anotar todas las variantes contempladas en la herramienta cuyas posiciones se encuentren dentro de los exones descritos en el fichero BED del panel genético. Esto se realiza a través de un script de R de implementación propia que el usuario tendrá disponible para su posible ejecución con el fin de actualizar dicha información.

Respecto a las bases de datos integradas en los anotadores, se pueden diferenciar tres grandes grupos que definen por tanto tres tipos de anotación diferentes.

En primer lugar, se tendrán en cuenta una serie de bases de datos destinadas facilitar la **frecuencia** de las variantes en la población. Estas son **ExaC** [81, 82], **dbSNP** [48, 49], **GnomAD** [50, 51], **1000 genomes** [83] y **Kaviar** [84, 85], permitiendo saber si una variante se trata de un simple polimorfismo o si por el contrario es una mutación genética de inusual aparición.

En segundo lugar, se encuentran las denominadas **In Silico Prediction Tools**, las cuales se encargan de predecir la patogenicidad de las variantes en función del posible cambio de aminoácido que puedan acarrear. Entre todas ellas, el flujo integra **SIFT** [86], **Polyphen2** [87], **MutationTaster** [88], **CHASM** [89] y **VEST4** [90].

Por último, están las bases de datos especializadas en **cáncer**, siendo de interés los identificadores asociados a cada variante obtenida. Esto, junto con la posterior **asignación de enlaces**, permite al usuario acceder de forma rápida a la información disponible en dichas bases de datos sobre las mutaciones detectadas. En el script se implementan **COSMIC** [91, 92], **ClinVar** [93], **CIViC** [94], **OncoKB** [95, 96] y **Varsome** [97].

El motivo por el que se han empleado tres anotadores distintos se debe al simple hecho de que no existe ninguna herramienta que logre contemplar por si sola todas las bases de datos mencionadas. Por tanto, se hará uso de todas ellas para unir posteriormente sus resultados en

un solo archivo XLSX.

Además de la información recabada en las bases de datos, es importante que el usuario conozca la seguridad con la que el flujo ha logrado identificar cada variante. Es por esto, que el script encargado de unir los resultados de todos los anotadores e insertar enlaces para los distintos identificadores, también consulta a los distintos archivos VCF generados por cada combinación alineador-variant_caller en búsqueda de los valores de **frecuencia alélica** (AF), **profundidad de lectura** (DP) y **Strand Bias** (SB) que se habían perdido en fases anteriores.

2.3.6. Fase 6 (DNA): Detección de CNAs

Retomando los archivos BAM correspondientes a las lecturas mapeadas y procesadas, el script hace uso del fichero procedente del mapeador que el usuario indique como preferente para llevar a cabo la detección de CNAs. El motivo por el que en este caso no se utiliza la información proporcionada por el resto de mapeadores se debe a que la identificación de CNAs es un proceso mucho menos sensible que el de SNVs e INDELs, por lo que usar varios alineadores en este caso se considera un exceso de información que incrementaría el tiempo de ejecución de forma innecesaria. Sin embargo, dado que cada herramienta usa su propio algoritmo en esta tarea, sí que se contempla nuevamente el uso de varios detectores para obtener los resultados más completos y fiables posibles.

En este proceso se recomienda el uso de las lecturas mapeadas correspondientes al tejido sano para poder construir una **referencia** sobre la que trabajar. Es por esto, que surge el primer problema. Como ya se ha mencionado anteriormente, el laboratorio prescinde de dicha información para llevar a cabo sus análisis, por lo que la situación ideal queda claramente descartada.

La solución más fiable, consiste en tomar el resto de archivos BAM de las muestras tumorales de los demás pacientes de la misma carrera para generar dicha referencia. Sin embargo, esto deja paso al siguiente problema. Para poder construir una referencia se necesita un mínimo de dos muestras, por lo que si el número de pacientes es menor que 3, esta solución tampoco es viable.

Por tanto, el flujo queda implementado de tal manera que, si el número de pacientes es mayor o igual que 3, se utilizan herramientas que cogen como referencia el resto de muestras, y en cualquier otro caso, se utilizan herramientas que no hagan uso de ninguna referencia para detectar los CNAs.

Para la primera situación se revisaron diferentes artículos que tratasen de comparar los resultados de diversas herramientas [98, 99, 100], de cuya lectura fueron seleccionadas **CnvKit** [101, 102], **CoNVaDING** [103, 104], **ExomeDepth** [105] y **Panelcn.Mops** [106, 107], mientras que en el segundo caso, se implementaron las únicas medianamente fiables que permitiesen llevar a cabo el proceso sin el uso de ninguna referencia, es decir, **CnvKit** [101, 102] y **Control-Freec** [108, 109].

Cabe mencionar que los resultados de estas herramientas son bastante diversos respecto al tipo y estructura del fichero de salida. Algunos de ellos proporcionan sus resultados en archivos de texto plano mientras que otros hacen uso de formatos como TSV, CSV o XLS. Además, cada herramienta basa su detección en una clase de región diferente puesto que algunos cuantifican el número de copias por cromosoma, otros por genes e incluso otros por exones.

Es por esto, que nuevamente se implementó un script de R capaz de analizar los archivos resultantes de cada detector y volcar toda la información en un único fichero XLSX. Para ello, se dedica una línea a cada exón de tal forma que las herramientas que detecten CNAs en regiones de mayor rango expongan su valor en el exón de inicio y se rellenen con guiones el resto de exones que cubran la alteración detectada. Además, con el objetivo de facilitar la lectura al usuario, se le aplican a los resultados significativos el color verde para las copias y el rojo para las pérdidas, de tal forma que el fichero resultante tenga información tanto textual como visual.

2.3.7. Fase 7 (RNA): Detección de reordenamientos genéticos

En esta fase se trata de identificar el cuarto tipo de variación contemplado en el flujo de trabajo, esto es, las fusiones génicas. Esta clase de alteración estructural es identificada mediante el análisis de las muestras de RNA, las cuales son utilizadas por diversas herramientas

tras su procesamiento.

De todas las fases contempladas hasta el momento, esta es la que más se ve afectada por la modalidad de secuenciación realizada en el laboratorio. El motivo es que apenas existen herramientas especializadas en la detección de fusiones génicas destinadas a datos procedentes de secuenciación Ion Torrent, de hecho, la mayoría de ellas solo permiten la entrada de lecturas pareadas.

Es por esto, que en esta fase del flujo se han intentado adaptar los detectores de mayor prestigio a las características específicas de los datos del laboratorio. Para ello, se realizó la recopilación bibliográfica correspondiente [110, 111] y se decidió integrar **Arriba** [112, 113], **FusionCatcher** [114] y **STAR-Fusion** [115], ya que obtienen los mejores resultados y a pesar de estar principalmente diseñados para el análisis de lecturas pareadas, permiten la entrada de datos de lectura única.

Al igual que siempre, los resultados de estas herramientas son unidos en un único archivo XLSX mediante la implementación propia de un script de R.

2.4. Herramientas integradas

A continuación, se detalla un análisis más exhaustivo del funcionamiento del script que permite determinar la correcta toma de decisiones y la correspondiente optimización de parámetros. En concreto, se expondrá información acerca de las herramientas que han sido asignadas a las fases más decisivas del flujo de trabajo.

2.4.1. Alineadores

Como ya se ha mencionado en la exposición de la fase 2, en el flujo de trabajo se integran un total de 8 mapeadores distintos. Su trabajo consiste en leer las lecturas procesadas de los archivos FASTQ y alinearlas sobre el genoma humano de referencia para obtener las lecturas mapeadas en el correspondiente fichero SAM.

- **BWA:** Burrows-Wheeler Aligner se define como un paquete software destinado al mapeo de **secuencias de baja divergencia** sobre un genoma de **referencia de gran tamaño**. Ambas especificaciones encajan con este estudio, por un lado las lecturas son bastante similares si se tiene en cuenta que la metodología de secuenciación del laboratorio hace uso de paneles de amplicones, y por otro lado al usar el genoma humano se está ante una clara referencia de grandes dimensiones. No obstante, esta herramienta consta de tres algoritmos: **BWA-backtrack**, **BWA-SW** y **BWA-MEM**. El primero de ellos se destina al mapeo de lecturas cortas de hasta 100pb propias de algunos equipos de secuenciación de Illumina, por lo que queda descartado. Los dos restantes, comparten características similares ya que ambos poseen soporte de lectura larga y secuenciación dividida, sin embargo, BWA-MEM se presenta como la mejor solución al ser la más rápida y proporcionar datos de mayor precisión y calidad. Es por esto, que se empleará este último algoritmo haciendo uso de sus parámetros básicos que permiten la proporción del archivo FASTQ con las lecturas procesadas y el índice previamente construido de hg19. **Versión:** 0.7.12-r1039

- **BOWTIE2:** Se presenta como una herramienta ultrarrápida y de memoria eficiente para alinear lecturas de secuenciación sobre referencias largas propias de mamíferos. Su implementación permite varios modos de alineación diferentes, sin embargo, de forma predeterminada y tal y como se recomienda para el caso abordado tras el procesamiento previo de las lecturas, se lleva a cabo un mapeo **end-to-end** en el que se pretende alinear bajo la consideración de todos los caracteres de las secuencias. Por tanto, BOWTIE2 será ejecutado con sus parámetros básicos que le proporcionan el índice y el archivo FASTQ correspondiente. **Versión:** 2.2.9

- **GMAP:** Esta herramienta integra al **mapeador GSNAP** (Genomic Short-read Nucleotide Alignment Program). Como su propio nombre indica, se especializa en el mapeo de **lecturas cortas**, ya sean pareadas o no, sobre genomas de referencia. A pesar de no estar del todo destinada al caso concreto de este proyecto debido a la longitud de lectura para el que se diseñó, el hecho de ofrecer **resultados de gran calidad** lo han convertido en una opción fiable dentro del flujo de trabajo. Como consecuencia de ser una herramienta poco flexible, apenas se permiten modificaciones en su ejecución, por lo que los

parámetros a utilizar son los usuales hasta el momento. **Versión:** v7 2015-12-31

- **Subread:** Esta herramienta se describe como un alineador de lectura de uso general que implementa una simple y elegante estrategia multi-semilla denominada **seed-and-vote**. Su funcionamiento se basa en el uso de varias semillas cortas llamadas subreads que son previamente extraídas de cada lectura para que todas ellas puedan "votar"sobre la ubicación que consideren óptima en el genoma. Tal y como expone su documentación, se trata de una herramienta adecuada a este estudio ya que su procedimiento favorece el alineamiento de lecturas largas. Además de los parámetros habituales, puesto que subread-align soporta tanto datos de RNA-Seq como de DNA-Seq, se le debe concretar el mapeo de lecturas procedentes de muestras de DNA mediante el uso del parámetro "-t" asignándole el valor 1. **Versión:** 2.0.0

- **HISAT2:** Consiste en un programa de alineación rápido y sensible para mapear lecturas procedentes de secuenciación NGS, tanto de ADN como de ARN, sobre genomas humanos de referencia. Para ello, utiliza un esquema de indexación denominado índice FM de gráfico jerárquico (HGFM), que se caracteriza por el **empleo de pequeños índices locales**. Esto, junto con diversas estrategias de mapeo consiguen una alineación rápida y precisa de las lecturas. Respecto a los parámetros, se utilizan los habituales hasta el momento. **Versión:** 2.1.0

- **NovoAlign:** Este alineador se define como una potente herramienta diseñada para mapear lecturas procedentes de **plataformas NGS** como Illumina, Ion Torrent y 454. La última versión gratuita disponible es V3.09.05 y a pesar de tener bastantes limitaciones respecto a la versión de pago, sigue siendo una opción considerable para el flujo de trabajo. Para llevar a cabo el alineamiento NovoAlign utiliza cualidades base y penalizaciones por espacios afines con el fin de encontrar la ubicación más probable de las lecturas. Se trata de un procedimiento lento pero seguro que suele obtener puntajes de calidad elevados. **Versión:** V3.09.05

- **GEM3:** Se trata de una herramienta de **mapeo de alto rendimiento** para alinear lecturas secuenciadas sobre genomas de referencia grandes como el genoma humano. A pesar de destinarse al mapeo de secuencias más largas que las procedentes de Ion Torrent, su

calidad y rapidez han demostrado ser eficientes con las muestras del laboratorio. Su procedimiento se basa en la indexación del genoma de referencia mediante un diseño de índice FM personalizado para posteriormente llevar a cabo una búsqueda adaptativa con huecos en función de las características de la entrada. Los parámetros son los habituales para la proporción de los archivos FASTQ y el correspondiente índice del genoma. **Versión:** v3.6.1

- **Kart:** Se considera un alineador de lectura NGS ultra-eficiente. Su implementación se basa en la **estrategia divide-and-conquer** que separa a las lecturas en regiones de dos tipos: las que son fáciles de alinear y las que requieren de mapeo con espacios. Tras ello, la herramienta se encarga de analizar cada región de forma independiente y componer el alineamiento final. Se trata de una herramienta rápida y eficaz cuya integración en el flujo se considera adecuada. **Versión:** v2.5.6

2.4.2. Llamadores de variantes

Los variant callers son las herramientas especializadas en la detectción de SNVs e INDELs. En el script se integran un total de 6 de ellas, cada cual con sus estrategias y algoritmos que se explican a continuación:

- **GATK Mutect2:** Esta herramienta se especializa en la llamada de mutaciones somáticas cortas mediante el uso de un **ensamblaje local de haplotipos** y la implementación de un **modelo de genotipado somático bayesiano** que difiere del MuTect original [116]. Dado que GATK integra la ejecución de esta herramienta en uno de sus "Best Practices Workflows" para la detección de variantes somáticas cortas [117], se ha seguido el procedimiento recomendado en dicho tutorial para incluir parte de su flujo de trabajo en el de este proyecto. Es por esto, que tras la llamada de variantes con Mutect2 se utilizan **LearnReadOrientationModel** [118], **GetPileupSummaries** [119], **CalculateContamination** [120] y **FilterMutectCalls** [121] con el objetivo de mejorar los resultados. Respecto a los parámetros, se han empleado los recomendados para el caso abordado en el laboratorio, esto es, la modalidad de solo tumor. **Versión:** 4.1.2

- **GATK Haplotypecaller:** Este variant caller se considera capaz de llamar a SNVs e INDELs simultáneamente a través del **ensamblaje de novo local de haplotipos** en una región activa. Esto significa que siempre que el algoritmo encuentra una región con signos de variación, descarta la información de mapeo existente y vuelve a ensamblar de nuevo las lecturas de esa zona. Al igual que la herramienta anterior, GATK expone el uso de esta herramienta en uno de sus apartados de mejores prácticas, en este caso para la detección de variantes germinales [122]. De forma análoga a la anterior, se siguen las recomendaciones fijadas en dicho tutorial para integrar el uso de esta herramienta en el flujo de trabajo. Por ello, su ejecución vendrá seguida de la herramienta **GenotypeGVCFs** [123], la cual devolverá el archivo VCF final. **Versión:** 4.1.2

- **VarScan2:** Su implementación se caracteriza por emplear un **enfoque heurístico-estadístico sólido** con el objetivo de llamar a las variantes que cumplan los umbrales deseados de profundidad de lectura, calidad de base, frecuencia alélica y significación estadística. Con ello, pretende distanciarse de la mayoría de variant callers que al igual que los anteriores se nutren del empleo de marcos probabilísticos como las estadísticas bayesianas, los cuales a pesar de obtener buenos resultados se ven ocasionalmente confusos ante ciertos factores como las muestras agrupadas, o aquellas que se encuentren contaminadas e impuras. Para su correcto funcionamiento requiere del uso previo de **samtools mpileup** sobre los archivos BAM recalibrados, para posteriormente poder llamar de forma independiente a los SNVs e INDELs mediante comandos distintos. Es por esto, que para proporcionar el VCF final con todas las variantes juntas se ha integrado en su ejecución el uso de **bcftools concat**, la cual analizará los resultados de VarScan2 y los unirá en un único fichero de salida. Respecto a los umbrales de calidad que se necesitan especificar, se ha decidido ser bastante permisivos debido a que posteriormente el script de R encargado de unir todos los resultados hace uso de su propio filtrado. **Versión:** 2.4.3

- **VarDict:** Es considerado un llamador de variantes **ultrasensible** destinado a la identificación de variantes sobre muestras tanto individuales como emparejadas en formato BAM. Se define como una herramienta **útil en la investigación del cáncer** que permite el reconocimiento de amplicones en secuenciación dirigida, lo cual se adapta por completo a la técnica se secuenciación elaborada en el laboratorio. Su principal inconve-

niente es la numerosa cantidad de variantes que detecta a causa de su ultrasensibilidad. Para solucionar este problema, la herramienta ya integra de forma predeterminada ciertos filtros basados en recomendaciones de usuarios [124]. Además, a modo de refuerzo no se debe olvidar que el flujo promueve la eliminación de falsos positivos mediante las estrategias de intersecciones y uniones mencionadas en la fase 4 así como a través del filtrado implementado en el script final de R. Para optimizar el tiempo de ejecución, el flujo emplea el **puerto de Java** de la herramienta original, ya que tal y como afirma su documentación, es 10 veces más rápida que la versión inicial de Perl. Además, para proporcionar el resultado en el formato deseado se hace uso del script **var2vcf_valid.pl** y de la herramienta **bcftools annotate** que permite eliminar ciertas etiquetas que causaban algunos conflictos durante el consensuado de resultados. **Versión:** v1.8.2

- **Freebayes:** Consiste en un detector de variantes genéticas **bayesiano** diseñado para encontrar pequeños polimorfismos e INDELs entre otros tipos de alteraciones. Freebayes se basa en **haplotipos**, de tal forma que llama variantes atendiendo a las secuencias literales de las lecturas y no tanto a su alineación precisa. Es por esto que la herramienta requiere nuevamente del archivo FASTA que contiene al genoma de referencia para llevar a cabo su funcionamiento. Para proporcionar el resultado final, es necesario aplicar una ordenación adicional a través de **vcf-sort** y la habitual normalización con **bcftools norm**. **Versión:** 1.3.2

- **LoFreq:** Es un llamador de variantes rápido y sensible para inferir SNVs e INDELs a partir de datos procedentes de secuenciación NGS. Se distingue del resto de variant callers en que hace un uso completo de los **puntajes de calidad de las bases** y otras fuentes de errores inherentes a la secuenciación, ya que son datos que los demás ignoran o utilizan para simples operaciones de filtrado. Afirma tolerar datos procedentes de **cualquier plataforma** ya que no emplea umbrales dependientes de la tecnología de secuenciación o de la máquina. Respecto al formato de salida, dado que LoFreq no llama genotipos no incluye las columnas SAMPLE y FORMAT en sus archivos VCF. Aunque estas columnas sean realmente opcionales en el formato estricto VCF, muchas de las herramientas integradas en el flujo requieren su existencia para tareas posteriores. Para solucionar esto, se ha empleado y modificado manualmente el script **lofreq2_add_sample.py** propor-

cionado por la propia herramienta para incluir las columnas que faltan con datos artificiales. Estos datos no son del todo inventados, ya que gracias a la reimplementación realizada sobre el script se añaden valores coherentes respecto al resto de información de las alteraciones. **Versión:** 2.1.3.1

2.4.3. Anotadores

- **ANNOVAR:** Se define como una herramienta de software eficiente que emplea información actualizada para anotar funcionalmente variantes genéticas detectadas de diversos genomas, entre ellos el humano (hg19). Posee tres clases de anotación diferentes, sin embargo, solo resulta de interés aquella basada en filtros que permite buscar las alteraciones en diversas bases de datos y extraer información de las mismas. Para ello, ANNOVAR solo necesita una lista de cuatro columnas: cromosoma, posición inicial, posición final, nucleótido de referencia y nucleótidos observados. Para transformar el archivo VCF final en dicha lista, se emplea el script **convert2annovar.pl** suministrado por la propia herramienta. Además, para llevar a cabo la anotación con la última versión de **COSMIC** (v.92 para GRCh37) se realizó un proceso adicional previo mediante el script **prepare_annovar_user.pl** que permitió preparar la librería tras la descarga de los correspondientes archivos de la base de datos. **Versión:** 2020-06-07

- **Open Cravat:** Esta herramienta es la actualización del original anotador **CRAVAT** que se define como un nuevo sistema de soporte de decisiones escalable y de código abierto para admitir la priorización de variantes y genes. Aunque la herramienta es conocida por su aplicación web, en el flujo de trabajo se ha integrado el correspondiente paquete de Python de la misma herramienta. Este se encarga de realizar la interpretación de variantes genómicas, incluido el impacto, la anotación y la puntuación de las variantes. Además, su implementación facilita la incorporación al script ya que además de la habitual exportación de resultados en excel, se permite la generación de un RData que contiene un dataframe con toda la información pertinente. Esto sin duda, resulta bastante cómodo para la posterior ejecución del script de R encargado de unir todos los resultados. **Versión:** 2.2.3

Figura 4: Se muestran las bases de datos consultadas por cada anotador. En primer lugar, los recuadros coloreados de verde indican que el anotador incluye a la base de datos correspondiente en su ejecución. Por otro lado, los amarillos hacen referencia a aquellos casos en los que a pesar de estar disponible la anotación, la información devuelta por la herramienta no se considera útil. Y por último, los recuadros rojos se destinan a las situaciones incompatibles entre las bases de datos y los anotadores.

- **MyVariant:** Este anotador proporciona servicios web REST fáciles de usar con el fin de consultar y recuperar datos de anotación de variantes procedentes de diversos recursos de datos populares. En su documentación, enfatiza en su diseño basado en la simplicidad y el rendimiento, que permite el mantenimiento de información fiable y actualizada. En el flujo de trabajo se integra el paquete de R asociado a esta herramienta, que tal y como se detalló en la fase 5, fue necesario realizar ciertas modificaciones en su empleo a causa de sus **necesidades de conexión** durante la ejecución del script. **Versión:** 1.22.0

Además de las características ya mencionadas de cada anotador, es importante mencionar qué **bases de datos** consulta cada uno para poder conocer con detalle el origen de la información reportada. Tal y como se muestra en la figura 4, gracias a la colaboración de las tres herramientas se puede cubrir la consulta de todas las bases de datos mencionadas hasta ahora.

Respecto al último recuadro naranja posicionado como un cuarto anotador, hace referencia a la asignación de enlaces mediante la obtención del **código HGVS** [125] de las variantes. Esto permite consultar a las bases de datos Varsome y OncoKB de forma directa.

2.4.4. Detectores de CNAs

Los siguientes tipos de herramientas son los detectores de alteraciones en el número de copias, de los cuales solo se expondrán aquellos que se emplean en el caso habitual del flujo cuando el número de muestras de entrada es superior a 3 y por tanto se consigue construir una referencia a partir del resto de lecturas mapeadas.

- **CnvKit:** Se presenta como una librería de Python que engloba una serie de scripts para inferir y visualizar el número de copias a partir de datos de secuenciación de ADN de alto rendimiento. Afirma soportar datos procedentes tanto de la plataforma Illumina como de Ion Torrent, proporcionando además una modalidad de ejecución propia para la **secuenciación con amplicones**. Sin embargo, tal y como recoge su documentación [126] esta especialización de la herramienta impide obtener información acerca del número de copias entre las regiones específicas, ofreciendo un solo valor para cada cromosoma. Además del correspondiente fichero XLS con los datos obtenidos, la herramienta permite la generación de dos **diagramas** en formato PDF que facilitan la visualización de los resultados. Su ejecución es común para todas las muestras ya que coge como entrada todos los archivos BAM y calcula el número de copias de todos ellos teniendo en cuenta en cada caso los datos de los demás. **Versión:** 0.9.6

- **CoNVaDING:** Su nombre proviene de las siglas de Copy Number Variation Detection In Next-generation sequencing Gene panels, lo cual define a la perfección el objetivo de esta herramienta. Su funcionamiento se basa en la entrada de varios archivos BAM para seleccionar las muestras con el patrón general más similar y así poder asignarlas como muestras de control. Tras ello, estas son empleadas durante la normalización de la profundidad de lectura en todos los objetivos para finalmente ofrecer una salida dividida en **tres niveles** de mayor a menor dureza de filtrado, quedándonos en este caso con el nivel intermedio. Al igual que en el caso anterior, solo es necesario llevar a cabo una ejecución común para todas las muestras. **Versión:** 1.2.1

- **ExomeDepth:** Consiste en un paquete de R que trata de llevar a cabo la detección de CNAs mediante el análisis de los datos asociados a la profundidad de lectura de las

muestras. Para su usual ejecución, la herramienta requiere de un conjunto de muestras de control y una de prueba sobre la que se quiere realizar el estudio. Para su integración en el flujo de trabajo se ha dividido su funcionamiento en dos scripts. En primer lugar, **exome_depth_counts.R** es ejecutada una única vez con la finalidad de construir la matriz de cuentas de todas las muestras y guardarla en un fichero RData. A continuación, **exome_depth_run.R** es ejecutada para cada paciente con el fin de que lo identifique en la matriz contenida en el RData para así considerar su muestra como la de prueba y el resto asignarlas automáticamente como las de control. Su salida expone la detección del número de copias para regiones significativas de diversos tamaños. **Versión:** 1.1.15

- **Panelcn.Mops:** Se trata de un paquete de Bioconductor cuya finalidad es la detección del numero de copias sobre datos procedentes de secuenciación NGS dirigida mediante paneles genéticos, lo cual se adapta a la perfección con este proyecto. Su funcionamiento es bastante similar al de la herramienta anterior, por lo que nuevamente se dividió su codificación en un script para la parte común y otro para la ejecución personalizada de cada muestra. El primero de ellos es **panelcnmops_counts.R** y al igual que antes se encarga de la construcción de la matriz de cuentas de todas las muestras. En segundo lugar, **panelcnmops_run.R** es ejecutado para cada paciente considerándolo como la muestra de prueba y dejando al resto como muestras de control. En este caso, la herramienta devuelve sin excepción un valor del número de copias para cada exón del panel genético suministrado por el laboratorio. **Versión:** 1.14.0

2.4.5. Detectores de reordenamientos genéticos

A continuación, se exponen las características y el proceso de integración de las herramientas especializadas en la detección de Fusiones Génicas. Como ya se comentó anteriormente, apenas existen detectores destinados a la modalidad de secuenciación realizada en el laboratorio, por lo que la mayoría de ellos fueron diseñados para estudios similares que disciernen en cierta medida de este proyecto. No obstante, han sido integrados en el flujo de trabajo bajo la asignación de aquellos parámetros que mejor puedan adaptarse a este estudio.

- **STAR-Fusion:** Esta herramienta es un componente de Trinity Cancer Transcriptome Analysis Toolkit (CTAT) [127] y emplea el alineador **STAR** para identificar fusiones génicas candidatas. Su ejecución a pesar de estar destinada a datos procedentes de la plataforma Illumina, soporta la entrada de archivos FASTQ de lectura única, lo cual se aprovechará para poder utilizarla en el script. Para su correcto funcionamiento, requiere de la descarga de una **librería de recursos** ligada a Trinity que contiene archivos de referencia del genoma así como ficheros de anotación **Gencode** [128]. Su salida se encuentra en formato TSV y expone para cada fusión candidata los genes implicados, los puntos de corte y el número de lecturas que respaldan la alteración estructural, entre otros datos. **Versión:** v1.10.0

- **FusionCatcher:** Su implementación se centra en la búsqueda de fusiones génicas somáticas nuevas o conocidas, translocaciones y quimeras sobre datos de RNA-Seq procedentes de cualquier plataforma NGS de Illumina (Solexa, HiSeq, NextSeq, MiSeq, MiniSeq). Al igual que con la herramienta anterior, se aprovechará el hecho de permitir la entrada de archivos FASTQ de lectura única para integrar su uso en el flujo de trabajo. Para ello, se utilizará el parámetro –single-end acompañado del archivo FASTQ precedido por -i. Durante su instalación, se recomienda la descarga de los archivos de referencia necesarios, para los cuales se proporcionan una serie de comandos tipo wget que instala por completo la librería pertinente asociada al genoma humano. En su tarea de detectar alteraciones estructurales, se nutre del alineamiento producido por **tres mapeadores**: **BOWTIE**, **BLAT** y **STAR**, reportando finalmente para cada fusión los datos habituales de posición y confianza más la lista de alineadores que han permitido su identificación. Esta herramienta por tanto, utiliza la misma estrategia elaborada en el flujo de trabajo para la detección de SNVs e INDELs, es decir, utilizar varios mapeadores con el fin de dar una respuesta fiable. **Versión:** 1.33

- **Arriba:** Se define como una herramienta de línea de comandos para la detección de fusiones de genes a partir de datos de RNA-Seq. Dado que fue desarrollada para su uso en **entornos de investigación clínica**, se construyó bajo un diseño centrado en obtener tiempos de ejecución cortos y alta sensibilidad. Su integración en el flujo se debe a que no se destina a ninguna plataforma de secuenciación concreta y que fue ganadora

del **DREAM SMC-RNA Challenge** [129], una competencia internacional para determinar el estándar de oro actual en la detección de fusiones génicas a partir de datos de RNA-Seq. A diferencia de los detectores anteriores, la fase de mapeo no se encuentra integrada en su implementación. No obstante, recomienda proporcionar como entrada el resultado del mapeador **STAR**, especificando incluso los parámetros óptimos para su ejecución y posterior inclusión en su funcionamiento. La salida se ofrece en formato TSV con la lista de las alteraciones estructurales detectadas, especificando para cada una de ellas los habituales puntos de ruptura y niveles de confianza. Además, la herramienta proporciona el script **draw_fusions.R**, el cual permite la generación de un PDF con la representación gráfica de dichas alteraciones que consigue mejorar la visualización de los resultados. **Versión:** v2.1.0

2.5. Scripts para la unión de resultados

Tras exponer la información más importante de cada una de las herramientas, se debe comentar con detalle la forma en que han sido unidos sus respectivos resultados para la formación de los archivos XLSX finales.

2.5.1. SNVs e INDELs: process_annotation.R:

Este script se encarga de proporcionar la salida del proceso asociado a la detección de SNVs e INDELs. Para su ejecución se requiere: la ruta de la carpeta del paciente destinada a volcar los resultados de los distintos variant callers, el nombre del fichero de salida, el archivo RData con las variantes anotadas por Open Cravat, el CSV procedente de la anotación llevada a cabo por ANNOVAR, la tabla de variantes contenida en el RData de MyVariant, el número de threads y por último la lista de variant callers empleados y establecidos en el fichero de configuración.

La ejecución comienza con la extracción de los datos asociados a la **frecuencia alélica**, **profundidad de lectura** y **strand bias**, contenidos en todos los archivos VCF generados por las distintas combinaciones mapeador-variant_caller. Dado que cada llamador de variantes

ofrece una estructura diferente para especificar los valores mencionados, se empleará una función distinta para cada uno de ellos.

Con el fin de optimizar el tiempo de ejecución, estas funciones serán ejecutadas de forma paralela sobre los VCF procedentes de los distintos mapeadores gracias a la función **mclapply()**, cuyo resultado en forma de lista de dataframes será convertido a un único dataframe mediante la función **merge()**. Una vez obtenido un solo dataframe para cada variant caller, serán unidos en una lista que facilite su posterior consulta en tareas futuras.

El siguiente paso consiste en la carga de las variantes anotadas procedentes de Open Cravat, ANNOVAR y MyVariant, de tal forma que solo quede un dataframe para cada anotador cuyos nombres de columnas sean representativos y fáciles de consultar.

Tras esto, el script se centra en la construcción de la cabecera del archivo de salida, el cual depende por completo de los mapeadores y los llamadores de variantes ejecutados en el flujo de trabajo. Para facilitar la comprensión del archivo final, se diseña una cabecera de cuatro líneas dispuestas de menor a mayor nivel de especificidad indicando en todo momento el origen de los valores expuestos, es decir, el nombre del anotador o el del mapeador y el variant caller pertinente.

La construcción del cuerpo se basa en recorrer el dataframe construido con los datos de Open Cravat y para cada variante buscar información en las tablas de ANNOVAR y MyVariant, así como en la lista de dataframes de los variant callers. De nuevo, para optimizar esta tarea, se hará uso de la función mclapply() con el objetivo de aprovechar todos los threads disponibles.

Una vez hecho esto, la lista formada por vectores correspondientes a cada variante se unen en un único dataframe. Además, se lleva a cabo la distinción entre columnas con valores propios de la clase character y aquellas con valores numéricos para así asignarles el formato adecuado antes del filtrado.

El procesamiento de estas variantes ha sido diseñado bajo las especificaciones del laboratorio. En total, se han cumplido 5 peticiones cuyos detalles se exponen a continuación:

- **Petición 1:** Ordenación alfabética de las variantes en función de la columna asociada a **Sequence Ontology** (SO) [130] la cual permite clasificar la naturaleza de la alteración genética.

- **Petición 2:** Eliminación de aquellas variantes cuyos valores de frecuencia alélica medios para cada variant caller sean todos menor que 0.05.

- **Petición 3:** Eliminación de aquellas variantes cuyos valores de profundidad de lectura medios para cada variant caller sean todos menor que 250.

- **Petición 4:** Eliminación de aquellas variantes cuya columna Sequence Ontology indique el valor synonymous_variant.

- **Petición 5:** Mantener únicamente aquellas variantes que sean codificantes (coding = Yes) o que en la columna Sequence Ontology tengan el valor splice_site_variant.

Tras el filtrado de las variantes, comienza la escritura del archivo XLSX final gracias al paquete **openxlsx**. Durante esta tarea, se crean los enlaces correspondientes, se asignan los anchos de las celdas, se fijan los colores distintivos de la cabecera para su correcta división en secciones y se fijan las 4 primeras filas correspondientes a la cabecera para facilitar la manipulación del archivo en excel.

2.5.2. CNAs: merge_cnv_outputs.R:

Su ejecución tiene como objetivo otorgar un fichero XLSX que logre reunir para cada paciente los resultados procedentes de todos los detectores de CNAs utilizados en el flujo. Para ello, requiere la entrada de: la ruta de la carpeta del paciente destinada a volcar los resultados de los distintos detectores, el archivo BED con las regiones diana, el nombre del fichero de salida y la lista de detectores empleados.

A pesar de tener un procedimiento similar al script anterior, en este caso se realizan todas las fases simultáneamente para cada detector. Esto es, tanto la cabecera como el cuerpo se van

construyendo a medida que se analizan los ficheros, registrando además aquellas posiciones en las que se ha detectado una ganancia o una pérdida significativa.

Dado que los resultados son bastante diferentes respecto a estructura y formato, el script se encarga de rellenar con guiones aquellas detecciones que cubran más de un exón a la vez que se le asignan los colores verde y rojo a las ganancias y pérdidas detectadas respectivamente. Para cada herramienta las reglas son las siguientes:

- **CnvKit:** Serán pérdidas aquellos valores numéricos inferiores a 1 y ganancias los superiores a 3.
- **CoNVaDING:** Dado que sólo devuelve los resultados que la propia herramienta considera significativos mediante las etiquetas DUP y DEL, se respetará su clasificación de forma directa.
- **ExomeDepth:** Igual que la herramienta anterior solo que en este caso las etiquetas claves devueltas por la herramienta son duplication y deletion.
- **Panelcn.Mops:** Dado que devuelve un valor por exón, serán pérdidas los exones marcados con un valor inferior a 1 o aquellas regiones con tres o más exones seguidos con valor igual o inferior a 1. E igual con las ganancias, es decir, se marcarán aquellos exones con valores superiores a 3 o a las regiones con al menos tres exones seguidos de valores igual o superior a 3.

Finalmente, gracias a la librería **openxlsx** se exportan los resultados a un archivo XLSX con la correspondiente cabecera y el pertinente coloreado de celdas.

2.5.3. Reordenamientos genéticos: merge_gene_fusion_outputs.R:

El diseño de este script se destina a la unificación de los resultados proporcionados por todas las herramientas especializadas en la detección de fusiones génicas. En este caso el script solo necesitará la ruta de la carpeta del paciente destinada las fusiones génicas, el nombre del fichero de salida y la lista de detectores ejecutados.

El primer paso se centra en la lectura de todos los archivos de salida para obtener la lista de dataframes asociados a cada detector a la vez que se construye el vector definitivo de fusiones junto con los correspondientes identificadores de Gencode.

Tras esto, el script diseña una cabecera de tres líneas para el archivo final donde se diferencian cuatro grandes secciones: **Fusions**, **Breakpoints**, **Confidence** y **Annotation**. Durante la construcción del cuerpo, se recorren las distintas fusiones detectadas para extraer la información que cada herramienta pueda aportar respecto a las secciones comentadas.

Al igual que siempre, se vuelve a hacer uso de la librería **openxlsx** para la redacción del archivo final de salida y el pertinente coloreado de la cabecera.

2.6. Modificaciones debidas a Picasso

La **potencia** que suministra la **supercomputadora Picasso** es lo que hace posible la ejecución de todas las herramientas descritas en un tiempo sostenible para el diagnóstico clínico llevado a cabo en el laboratorio. No obstante, el uso del flujo de trabajo construido en este proyecto en la supercomputadora, requiere de una serie de adaptaciones y sacrificios.

En primer lugar, a causa de no tener **permisos de usuario**, la mayoría de las herramientas, librerías y versiones que no se encuentren integradas en su sistema deben ser **instaladas de manera local**. A pesar de que la supercomputadora dispone de un equipo técnico disponible para la resolución de problemas, no cubren la instalación de librerías y herramientas específicas que solo van a ser utilizadas por un único usuario.

En la figura 5, se muestra una tabla con las herramientas que se incluyen en Picasso y las que han debido ser instaladas localmente.

Además, la carga de las herramientas en Picasso a través de sentencias **"module load"** generan en muchas ocasiones incompatibilidades en las versiones de lenguajes básicos como Java, Python, Perl, C, etc. Es por esto, que a lo largo del script se llevan a cabo numerosas activaciones y desactivaciones de estas herramientas con el fin de evitar y esquivar los conflictos.

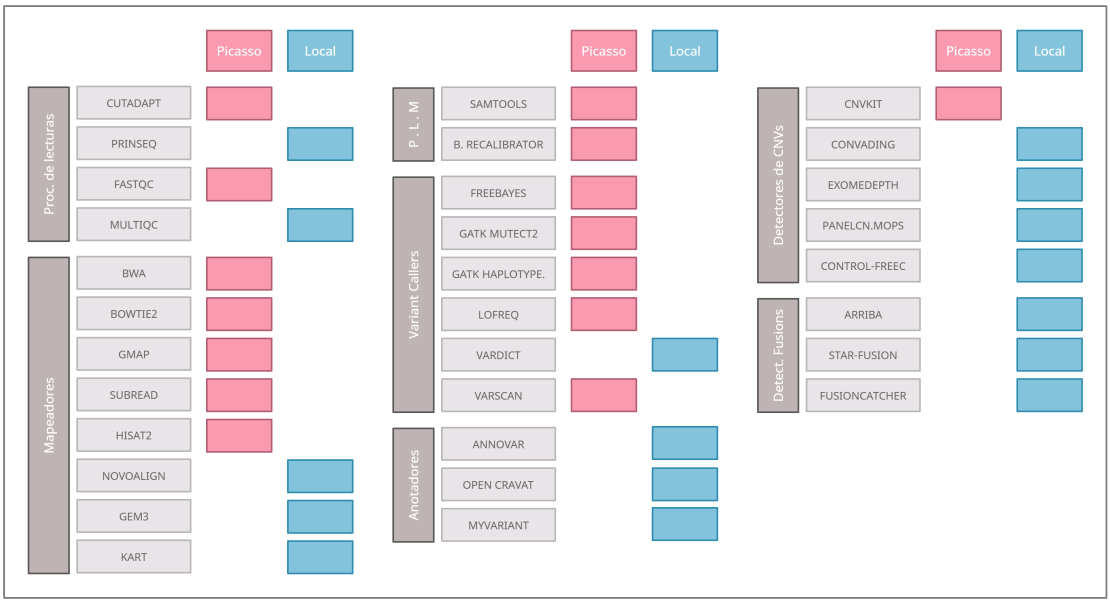


Figura 5: Las herramientas integradas en el flujo de trabajo se pueden dividir en 7 grupos: aquellas destinadas al procesamiento de lecturas iniciales, los mapeadores, las encargadas del procesamiento de las lecturas mapeadas, variant callers, anotadores, detectores de CNAs y detectores de fusiones génicas. Aquellas herramientas que poseen un recuadro rosa significa que se encuentran ya instaladas en la supercomputadora mientras que aquellas con un recuadro azul han sido instaladas localmente.

Por otro lado, la complejidad asociada a la descarga de librerías de Perl de forma local ha provocado la necesidad de incluir en el script la sobrescritura de **variables de sistema** como **PERL5LIB** o **LD_LIBRARY_PATH**, lo cual ha sido imprescindible para poder emplear CoNVaDING y STAR-Fusion.

Por último, la **falta de conexión a internet** durante la ejecución del flujo de trabajo han imposibilitado realizar un proceso de anotación permanentemente actualizado. Es por esto, que el usuario o el futuro bioinformático del laboratorio deberá mantener actualizados los archivos correspondientes a las distintas bases de datos empleadas.

2.7. Paralelización y multiprocesado

Para conseguir optimizar al máximo el tiempo de ejecución, resulta fundamental aprovechar todas las hebras disponibles en cada fase del flujo de trabajo. Puesto que el script permite

la entrada de un número indefinido de pacientes, se pretende que el flujo utilice al máximo la potencia suministrada por la supercomputadora independientemente de dicha cantidad, incluso si se incluye una única muestra. Además, por lo comentado anteriormente, se conoce que hay un punto de unión en la detección de CNAs donde todas las muestras sin excepción deben encontrarse al mismo nivel de ejecución. Es por ambas razones, que la paralelización estricta sobre diferentes muestras fue descartada.

La estrategia principal llevada a cabo en el script consiste en ejecutar las herramientas que así lo permitan en **modo multihebra** (lo cual aprovecha siempre el total de threads disponibles independientemente del número de muestras) y para aquellas herramientas que en dicha fase no dispongan de esa opción sean paralelizadas mediante su integración en una función y su llamada a través de **GNU Parallel**.

Por un lado, respecto a las herramientas que permiten multithreading con el uso del habitual parámetro –threads o -p, hay dos excepciones. Estas son Arriba y STAR-Fusion, las cuales a pesar de permitir esta modalidad de ejecución no emplean la totalidad de hebras debido a que producen un aumento excesivo de demanda de **memoria RAM** que acaba colapsando la ejecución. Es por esto, que se realizó para ambas herramientas una serie de ejecuciones a través de **/usr/bin/time -v**, con el fin de encontrar el valor más eficiente de su parámetro –threads que aprovechase el mayor número de hebras posible a la vez que no excediera la capacidad RAM disponible. Finalmente se le asignó 4 threads a la herramienta STAR-Fusion y 8 a Arriba.

Por otro lado, las herramientas que han requerido de GNU Parallel han sido GATK BaseRecalibrator durante el procesamiento de las lecturas mapeadas y diversos variant callers en la fase asociada a la detección de SNVs e INDELs.

A pesar de que Freebayes, LoFreq y GATK HaplotypeCaller soporten la ejecución multihebra, se descartó su uso debido a que Freebayes requería de la partición manual previa del archivo BAM de entrada, LoFreq no obtuvo apenas mejora en su tiempo de ejecución y **GATK HaplotypeCallerSpark** [131] advierte de ser una versión beta que no se hace responsable de cualquier modificación en los resultados. Después se encuentra VarDict, que gracias a su portal Java la ejecución multithreading funciona sin problema, y por otro lado están VarScan y

Gatk Mutect2 que no permiten esta modalidad de ejecución. Por tanto, el único variant caller que hace uso de la ejecución multihebra es VarDict mientras que el resto son integrados en una función que se llama de forma paralela con GNU Parallel.

Por último, hay una serie de herramientas que debido a su bajo coste computacional y al poco tiempo que necesitan para su funcionamiento, a pesar de no soportar la ejecución multihebra no han sido paralelizados de ninguna forma, por lo que son ejecutados bajo un único thread. Estas son por ejemplo MultiQC, Qualimap, algunos detectores de CNAs o los anotadores de variantes.

3

Resultados y Discusión

Las muestras utilizadas para llevar a cabo el testeo del script proceden del Laboratorio de Biología Molecular del Cáncer ubicado en el CIMES. Estas muestras se encuentran anonimizadas y pertenecen a pacientes con distintos tipos de cáncer, obteniendo archivos de entrada de diferentes tamaños y calidades de secuenciación.

El flujo de trabajo se ha puesto en marcha para un total de **7 pacientes** (7 muestras de DNA y 7 de RNA), lo cual será aproximadamente la mitad de su uso cotidiano en el laboratorio. Para ello, se seleccionaron los mapeadores **BWA**, **GEM3** y **HISAT2** junto con **todos los variant callers y detectores disponibles**.

En este apartado, se tratarán todos los aspectos asociados a la salida proporcionada por el flujo de trabajo tras esta ejecución, así como la **discusión** pertinente acerca de la eficiencia del script construido para su uso clínico rutinario.

3.1. Distribución de ficheros

Para facilitar la navegación del usuario sobre los resultados, se ha diseñado una distribución de ficheros de salida cómoda y útil que permita la búsqueda de información directa mediante la mera intuición. Al acabar la ejecución del script, el usuario tendrá **un directorio para cada paciente**, de tal forma que los datos incluidos en dicha carpeta solo posean relación con sus muestras.

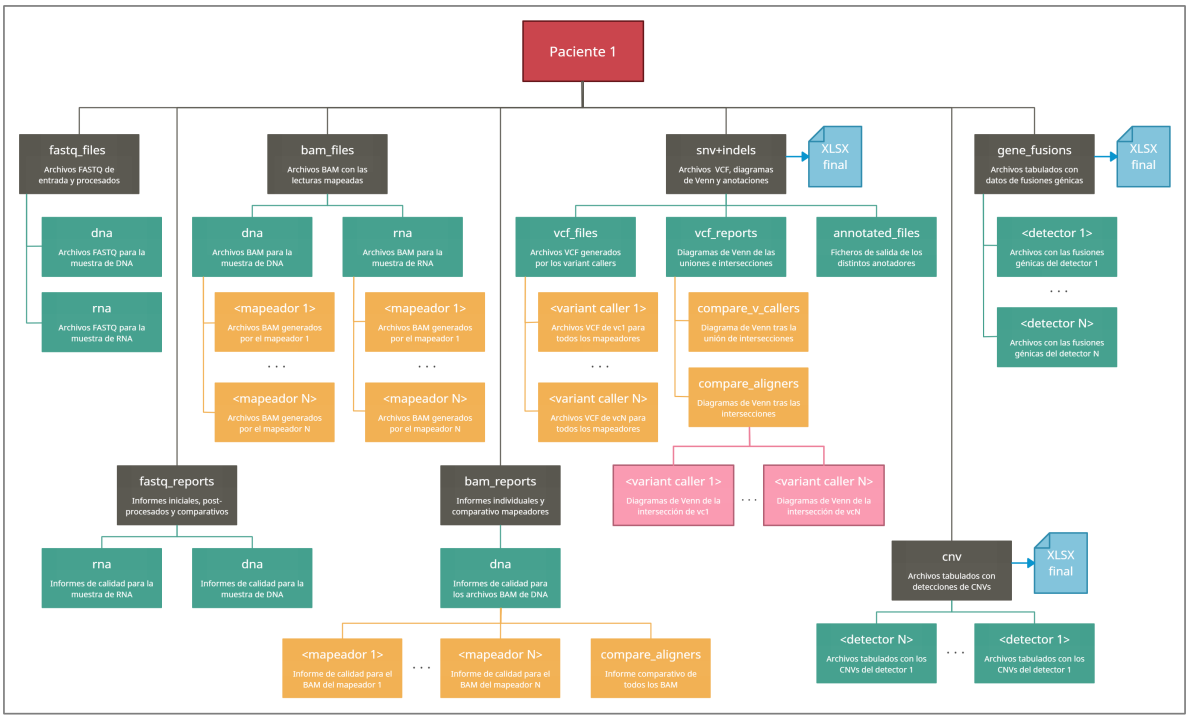


Figura 6: Diagrama que muestra la estructura del directorio generado para cada paciente tras la ejecución. De menor a mayor distancia respecto al directorio raíz representado de color rojo, se emplean los colores marrón, verde, naranja y rosa. Además, se muestran de color celeste los archivos XLSX finales con el fin de posicionarlos dentro de la distribución expuesta.

Dentro de estas carpetas, se distinguen 7 secciones principales divididas en subdirectorios:

- **fastq_files:** Este subdirectorio contiene los archivos FASTQ iniciales y procesados tanto de la muestra de DNA como la de RNA del paciente.
- **fastq_reports:** En esta sección se incluyen los informes de calidad iniciales, post-procesados y comparativos tanto para la muestra de DNA como la de RNA del paciente.
- **bam_files:** Tras su primera división entre DNA y RNA, en su interior se dispone un subdirectorio para cada mapeador que contiene el archivo BAM con las lecturas mapeadas por el mismo.
- **bam_reports:** Con una distribución similar a la anterior, en este caso la carpeta de cada mapeador contiene el informe de calidad asociado a su archivo BAM. Además, existe un subdirectorio adicional destinado para la inclusión del informe comparativo.

- **snv+indels:** En esta sección se volcará toda la información generada durante la detección de SNVs e INDELs. Esto es, los archivos VCF producidos por cada variant caller, los diagramas de Venn representativos de las intersecciones y uniones, los ficheros generados por los anotadores de variantes y por último el archivo XLSX final con todas las variantes detectadas.

- **cnv:** Esta sección contiene el archivo XLSX final con todos los CNAs y un subdirectorio destinado a cada detector, de tal forma que cada uno contenga los archivos tabulados proporcionados en su salida.

- **gene_fusions:** Igual que la sección anterior pero para el caso de las fusiones génicas.

Para poder visualizar con mayor detalle la distribución completa de directorios, se ha elaborado el diagrama expuesto en la figura 6 donde se representa la estructura de la carpeta asociada a un paciente.

3.2. Archivos intermedios generados

Resulta evidente que la información más relevante generada por el flujo de trabajo se encuentra en los archivos XLSX finales, los cuales incluso permiten conocer el origen, ya sea mapeador, detector o anotador, de los datos suministrados.

No obstante, si se pretende eliminar del todo el verdadero concepto de "caja negra", es necesario ser lo más transparente posible con el usuario. Es por esto, que además de las tablas finales, el usuario puede consultar si lo desea las salidas proporcionadas por las herramientas de mayor trascendencia en el proceso con el fin de que pueda analizar, entender y revisar el funcionamiento del flujo de trabajo ejecutado.

Entre estos archivos, se encuentran las lecturas procesadas por CUTADAPT y PRINSEQ, los BAM ya recalibrados procedentes de los distintos mapeadores seleccionados, los VCF de cada combinación entre variant callers y alineadores, los VCF resultantes de las intersecciones, el VCF con las variantes unidas de todas las intersecciones, las variantes anotadas por cada

anotador, los archivos tabulados de los detectores tanto de CNAs como de fusiones génicas, así como los informes de calidad, diagramas de Venn y el archivo de registro con los tiempos de ejecución de cada herramienta.

3.2.1. Informes de calidad

El flujo de trabajo genera de forma opcional informes de calidad tanto de los archivos FASTQ como de las lecturas mapeadas contenidas en los ficheros BAM ya recalibrados. Si se presta atención a las lecturas sin alinear, están por un lado los **informes generados por FastQC** antes y después de su procesamiento, y por otro el informe comparativo realizado por MultiQC donde se permite ver con mayor claridad las mejoras obtenidas.

Respecto a los informes generados por FastQC, la gráfica más significativa es la denominada "Per base sequence quality", la cual representa la calidad media de las bases a lo largo de las secuencias. Esta calidad suele decaer a medida que las bases se alejan del inicio de la lectura, por lo que la curva es generalmente de carácter descendente. No obstante, gracias al procesamiento, y en concreto, a técnicas como trimming integradas en la ejecución de PRINSEQ, se consiguen recortar los extremos de baja calidad en las secuencias. Esta técnica, también resulta empleada en otros estudios donde se verifica que el proceso de recorte realizado resulta beneficioso para posteriores tareas como el alineamiento, obteniendo una mayor proporción de lecturas correctamente mapeadas [132, 133, 134].

Esto, tal y como se muestra en la figura 7, consigue incrementar la calidad media de las lecturas y por tanto mejorar claramente la trayectoria de la curva en la gráfica. Como también se observa en dicha figura, el informe generado por MultiQC se encarga de superponer ambas curvas con el fin de ver con mayor detalle la mejora de calidad obtenida tras el procesamiento.

En relación a los archivos BAM, se encuentran los **informes de calidad realizados por Qualimap** acerca de las lecturas mapeadas por cada alineador, los cuales se generan bajo la consideración del archivo BED donde se exponen las coordenadas de las regiones secuenciadas. En los informes individuales, se generan varias tablas y gráficas cuya información más significativa se expone en la figura 8.

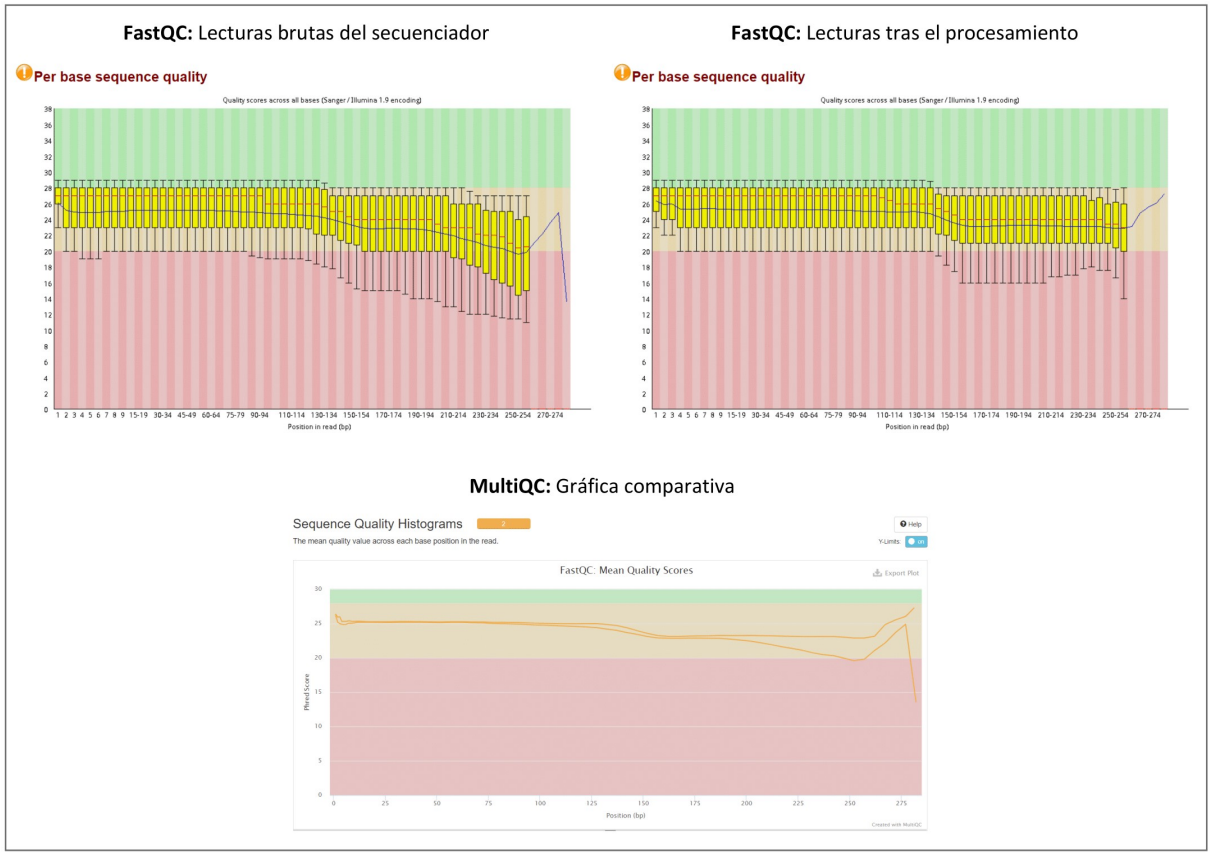


Figura 7: Gráficas de calidad de las bases a lo largo de las secuencias para las lecturas iniciales y procesadas. En la primera línea se muestran las gráficas correspondientes a FastQC y en la segunda la perteneciente al informe comparativo generado por MultiQC.

Como se puede ver, los informes reflejan datos sobre el porcentaje de lecturas mapeadas, media y desviación de cobertura, calidad de mapeo medio y el porcentaje de secuencias duplicadas, el cual siempre va a resultar alto si se tiene en cuenta la modalidad de secuenciación del laboratorio basada en amplicones.

Entre otras gráficas, destaca aquella que muestra la cobertura de mapeo a lo largo del genoma de referencia. Dado que los datos proporcionados por el laboratorio provienen de experimentos de secuenciación dirigida, es habitual encontrarnos con una curva irregular de cambios bruscos, donde la cobertura de secuenciación crece de forma radical en las zonas del panel genético. Este tipo de resultado también es apreciable en el artículo de Kenneth Day et al. [135] donde se grafica la profundidad de lectura a lo largo del cromosoma 11 tras una secuenciación dirigida.

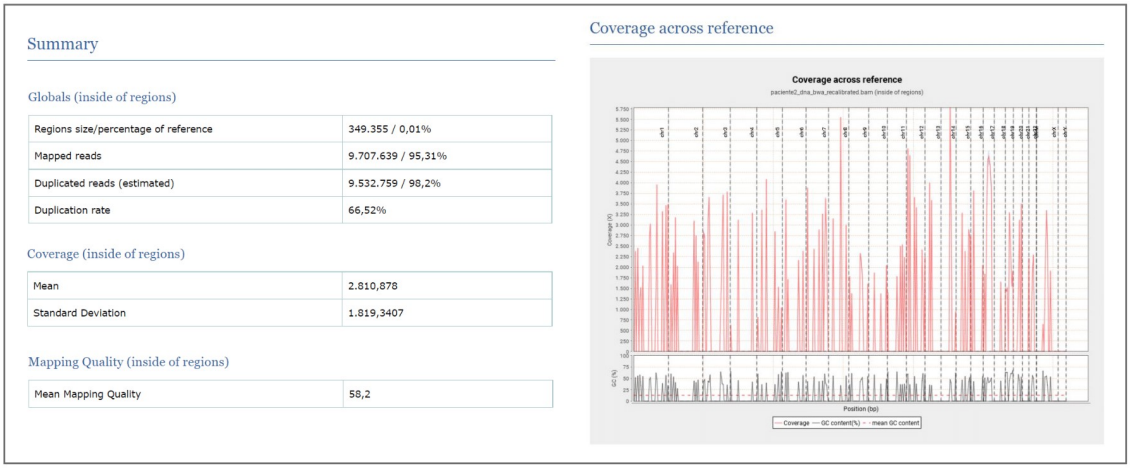
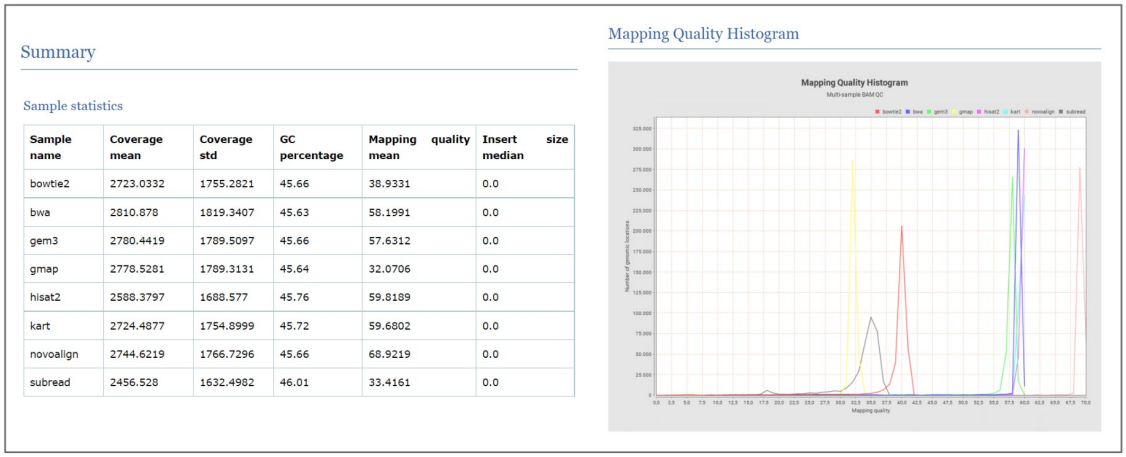


Figura 8: Se muestran capturas de uno de los informes individuales generados por Qualimap. A la izquierda se observan algunas tablas extraídas del apartado resumen del informe donde se especifican valores asociados a la cobertura y la calidad de mapeo. A la derecha se muestra la gráfica que trata de representar la profundidad de lectura a lo largo del genoma completo.

Aunque en la ejecución de testeo no se seleccionaron todos los alineadores, sí que en pruebas anteriores se utilizaron todos ellos con el fin de poder establecer un subconjunto de preferencia. Tras obtener y procesar todos los archivos BAM, se ejecutó la herramienta Qualimap para la generación de un **informe comparativo** que permitiese analizar los resultados de los distintos mapeadores. En la figura 9, se muestran los datos más significativos de este tipo de informes.

En la tabla de datos, dado que todas las lecturas mapeadas provienen de la misma muestra, es normal que los valores de cobertura y contenido GC sean similares. Sin embargo, interesa comparar la columna **Mapping quality mean** ya que brinda información acerca de cómo se han mapeado las lecturas, lo cual es repercusión directa del alineador utilizado. Como se puede ver, el mapeador que se distingue claramente del resto por su calidad de mapeo es Novoalign, dejando en segundo plano una serie de alineadores como BWA, HISAT2, GEM3 y KART. Esto también se visualiza gráficamente en la sección **Mapping Quality Histogram**, donde se representa el número de lecturas (eje y) en función de su calidad de mapeo (eje x) clasificadas por el alineador del que provienen (colores leyenda).

En dicha gráfica se puede apreciar el pico obtenido por Novoalign en calidades de mapeo superiores y la posterior aparición de BWA, HISAT2, GEM3 y KART con picos situados en



Summary

Sample statistics

Sample name	Coverage mean	Coverage std	GC percentage	Mapping quality mean	Insert size median
bowtie2	2723.0332	1755.2821	45.66	38.9331	0.0
bwa	2810.878	1819.3407	45.63	58.1991	0.0
gem3	2780.4419	1789.5097	45.66	57.6312	0.0
gmap	2778.5281	1789.3131	45.64	32.0706	0.0
hisat2	2588.3797	1688.577	45.76	59.8189	0.0
kart	2724.4877	1754.8999	45.72	59.6802	0.0
novoalign	2744.6219	1766.7296	45.66	68.9219	0.0
subread	2456.528	1632.4982	46.01	33.4161	0.0

Mapping Quality Histogram

Figura 9: Se expone parte del informe generado por Qualimap destinado a la comparación de todos los alineadores para una muestra concreta del laboratorio. A la izquierda se presenta una tabla con los valores promedios de calidad de mapeo y a la derecha se muestra el histograma asociado a dichas medias que permite profundizar un poco más en la comparación de herramientas.

torno al rango de calidad 55-60. Es por esto, que se establecieron estos 5 mapeadores como provisionalmente preferentes a la espera de contrastar sus respectivos tiempos de ejecución.

Este resultado era esperable si se tiene en cuenta los valores de precisión y especificidad obtenidos en algunos artículos referenciados anteriormente. Por ejemplo, el artículo de Hsin-Nan Lin et al. [44] que destaca los resultados obtenidos por el mapeador KART o el publicado por Subazini Thankaswamy-Kosalai et al. [29] donde se menciona la calidad proporcionada por BWA y NovoAlign. Además, existen otras publicaciones como la de Hanna Marie Schilbert et al. [136] que destaca la precisión de GEM3 y NovoAlign o la escrita por Brittney N. Keel et al. [137] donde se posiciona a BWA por encima de BOWTIE y HISAT2.

3.2.2. Diagramas de Venn

Otra información opcional que genera el script, son los diagramas de Venn asociados a los procesos de intersección y unión de variantes. En primer lugar, si se atiende a las **intersecciones**, se aprecia un diagrama de Venn para cada variant caller donde se comparan las variantes procedentes de distintos alineadores. Para un paciente aleatorio de la ejecución de testeo realizada, se obtienen los diagramas de Venn expuestos en la figura 10.

Como se observa, en todos los casos la mayor parte de las variantes detectadas son comunes a todos los alineadores ejecutados. Dichas alteraciones se consideran las más fiables para su mantenimiento en las siguientes fases. Por otro lado, existen una gran cantidad de variantes que han sido identificadas a través de dos mapeadores sin consentimiento de un tercero. Aunque estas variantes no sean del todo seguras, el hecho de que dos mapeadores hayan conducido a su detección no se considera una mera casualidad y por tanto pasan a incluirse en la lista definitiva. Por último, para todos los variant callers existen alteraciones que sólo han sido detectadas a través del alineamiento realizado por un único mapeador. Estas, suelen producirse por errores de alineamiento, y por tanto, al no ser fiables para un juicio clínico, son eliminadas del flujo de trabajo.

Si se atiende por ejemplo al caso de VarDict, se puede apreciar la importancia y trascendencia que tiene el uso de esta estrategia. Esto es, si en lugar de utilizar los tres mapeadores sólo se contemplase el alineador HISAT2, no sólo se habrían incluido 2093 probables falsos positivos, sino que además se hubiesen perdido 6570 variantes considerablemente fiables que HISAT2 ni siquiera contempla.

Si se analizan con detalle estos diagramas, no sólo permiten comparar alineadores, sino que también se puede saber la cantidad de variantes que llama cada herramienta, teniendo por tanto una idea estimada de su sensibilidad y su porcentaje de aportación a la lista definitiva de variantes. En este caso, se aprecia como los variant callers con un mayor número de alteraciones son GATK Mutect2 y VarDict, mientras que LoFreq se encuentra en el extremo opuesto siendo el que menos variantes detecta.

Este resultado es bastante coherente con lo expuesto en la bibliografía consultada. Como ya se ha mencionado, VarDict se trata de una herramienta ultrasensible definida por su propia documentación [63, 64]. Esta afirmación además se refleja en los resultados de otros papers como el de Xiaopeng Bian et al. [55] donde se comprueba que VarDict es la herramienta que posee de forma simultanea el mayor número de falsos y verdaderos positivos a causa de generar un gran volumen de alteraciones. Por otro lado, en otros artículos destinados a la comparación de variant callers como el de Qing Wang et al. [54], se puede apreciar como Mutect2 devuelve una cantidad de variantes muy superior al resto de herramientas.

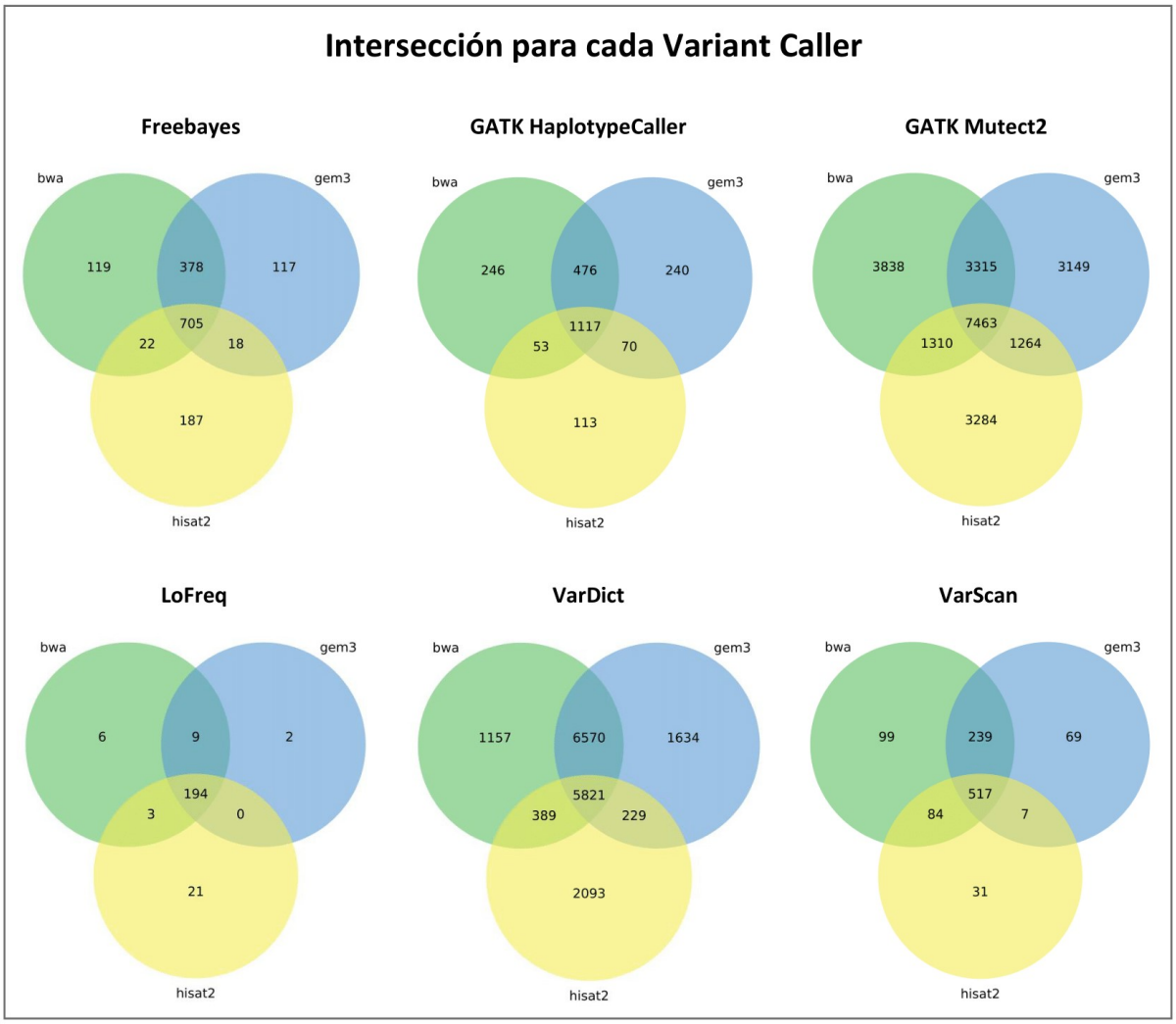


Figura 10: Se muestra un diagrama de Venn para cada variant caller comparando las variantes en función del mapeador de procedencia. Los diagramas son generados automáticamente por el flujo de trabajo para cada paciente a través de la herramienta vcftoolz.

Esto permite esbozar una idea del resultado más probable en el diagrama de unión. Esto es, VarDict y GATK Mutect2 aportarán un mayor numero de alteraciones delimitando un claro espacio de búsqueda propio, mientas que LoFreq al poseer pocas variantes apenas aportará casos que no hayan sido descubiertos por el resto de llamadores de variantes, haciendo prescindible su existencia en el flujo.

En la figura 11, se muestra el diagrama de Venn asociado al proceso de **unión de las intersecciones** expuestas anteriormente. Dado que hay 6 variant callers, el diagrama de Venn resultante se representa a través de triángulos y posee una mayor densidad de datos que en

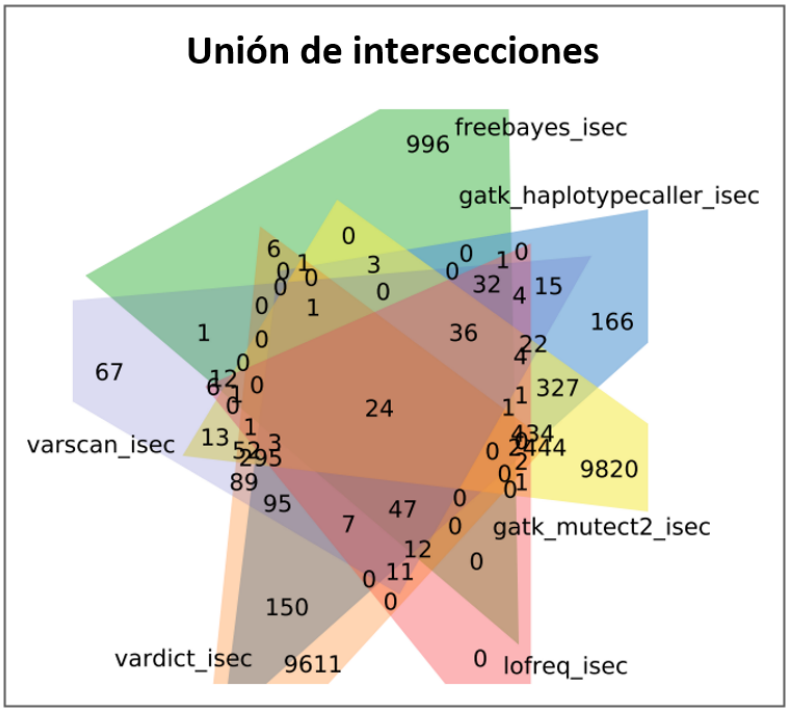


Figura 11: Representación del diagrama de Venn asociado a la unión de las intersecciones expuestas en la anterior figura. Al igual que en el caso anterior, este diagrama es automáticamente generado durante la ejecución del script.

los casos anteriores. Efectivamente, se confirma la anterior sospecha ya que se observa como la punta del triángulo propio de LoFreq tiene 0 variantes propias, mientras que GATK Mutect2 y VarDict son los que más variantes aportan a la lista.

Cabe mencionar que si se comparan ambas figuras, se puede apreciar que la similaridad entre variantes detectadas por igual llamador pero procedentes de distinto mapeador (figura 10) es mayor que en las variantes identificadas por diferentes variant callers (figura 11). De hecho, tal y como se muestra en el diagrama referente a la unión de intersecciones, sólo 24 variantes han sido detectadas por todos los variant callers.

Esta baja concordancia entre distintos llamadores de variantes, es un problema bastante tratado y estudiado en el ámbito de la bioinformática. Además de los artículos que se encargan de comparar este tipo de herramientas como el de Ze Kun Liu et al. [56] donde se exponen diversos diagramas de Venn que reflejan esta conducta, existen artículos destinados al tratamiento exclusivo de este asunto como es el caso del escrito por Jason O'Rawe et al. [138].

Es por esto, que el uso de varios mapeadores y su intersección permite aumentar la fiabilidad de las variantes para cada llamador, mientras que la unión de los resultados de múltiples variant callers aporta diversidad en los algoritmos de búsqueda y una ampliación del abanico de resultados.

3.3. Tablas finales

Tras la ejecución del script, el usuario posee para cada paciente tres archivos XLSX correspondientes a las tres secciones principales del flujo de trabajo: SNVs + INDELs, CNAs y Reordenamientos Genéticos.

3.3.1. SNVs + INDELs

En esta tabla, el usuario contemplará **una fila por variante** y un total de **143 columnas agrupadas por secciones** gracias al diseño sesgado en niveles de la cabecera. Para poder ilustrar el usual contenido de este archivo, se ha tomado como ejemplo una variante determinada de la misma muestra utilizada hasta el momento. Dada la dimensión de los datos, se han dividido sus 143 celdas en 4 figuras que se explicarán a continuación.

En primer lugar, para cada variante se muestra **información básica** acerca de su repercusión más directa tanto en el genoma, como en el transcriptoma y proteoma, en función de su posición y características. En el caso concreto expuesto en la figura 12, se informa de una alteración detectada en el chromosoma 17, en la posición 7579472, donde el nucleótido habitual G ha sido cambiado por una C. Además, se detalla que la variante de tipo "missense" se encuentra en una región codificante del gen TP53, que conlleva un cambio aminoacídico en el codón 72 donde una prolina pasa a ser una arginina. También se proporciona el ID del transcrito, el cambio en el DNA complementario y la alteración genética bajo nomenclatura HGVS.

Tras esto, las siguientes columnas de la tabla aportan información sobre la forma y seguridad con la que la variante ha sido detectada por los diferentes **variant callers** a través de todos

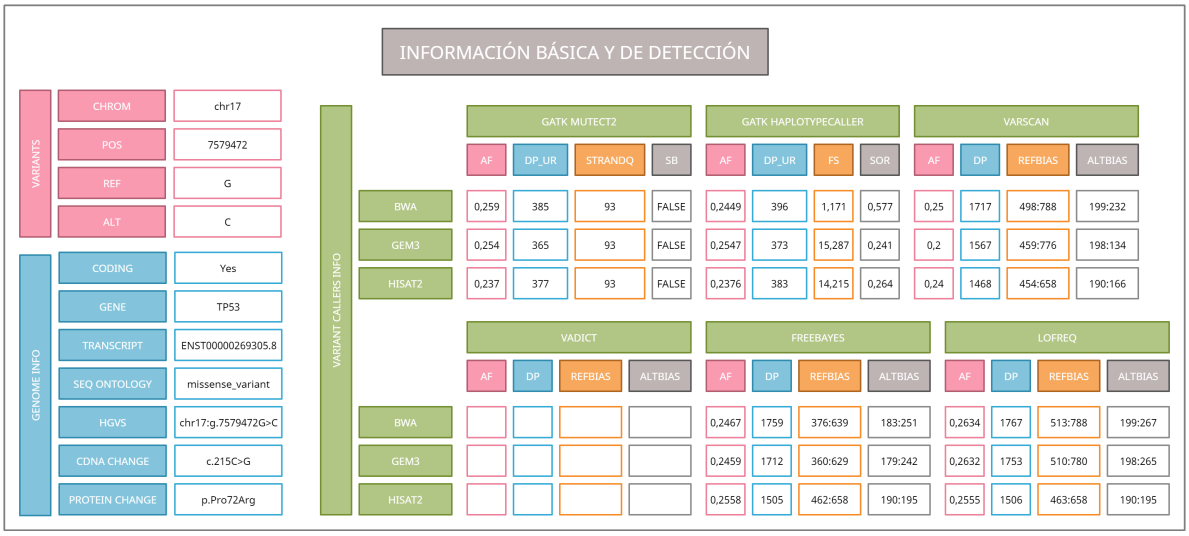


Figura 12: Primera parte del ejemplo de una variante contenida en la tabla final del archivo XLSX. Se muestra la información básica de la variante de color rosa y azul, y los datos asociados a su detección por las distintas combinaciones entre alineadores y variant callers de color verde. Además, se realiza una distinción de colores asignando el rosa para la frecuencia alélica, el azul para la profundidad de lectura y el naranja y gris para valores asociados al strand bias.

los mapeadores seleccionados. Esta información se divide en tres grandes grupos: **frecuencia alélica** (AF), **profundidad de lectura** (DP) y **strand bias** (REFBIAS, ALTBIAS).

- **Frecuencia alélica:** Permite conocer la proporción con la que la variante se ha detectado en la muestra. Su valor se calcula mediante la división del número de lecturas que contienen la alteración genética entre la profundidad de lectura alcanzada en su posición.
- **Profundidad de lectura:** Es el número total de lecturas mapeadas en la localización sobre la que se sitúa la variante.
- **Strand Bias:** Brinda información acerca del número de lecturas que contienen la alteración para cada hebra de la cadena de DNA. Si una alteración se produce en una hebra, por consecuencia directa debe expandirse a la otra. Por tanto, en caso de que el número de lecturas no sea similar para ambas, permite cuestionar la fiabilidad de la variante.

En el ejemplo, puede apreciarse que la variante ha sido identificada por todos los variant callers menos por VarDict. Los valores de frecuencia alélica son bastante similares ya que en

todos los casos se posiciona entre 0,2 y 0,3, lo cual significa que la variante se detecta de forma prácticamente unánime por todos los variant callers en un 20-30 % de las lecturas mapeadas en su posición.

Respecto a la profundidad de lectura, existen dos grupos claramente diferenciados. Por un lado se encuentran las herramientas procedentes de GATK, las cuales reportan profundidades de lectura referentes a secuencias filtradas (debido a los amplicones muchas las detecta como duplicados) [139], mientras que el resto proporcionan valores de DP totales que son lógicamente más altos y acordes a lo visto durante los informes de calidad generados por Qualimap.

En el ejemplo, puede apreciarse como la profundidad de lectura habitual es de en torno a 1500-1700 lecturas, mientras que los detectores de GATK lo reducen a unas 400 secuencias únicas. Este es un problema comentado en varios foros [140, 141] que a pesar de que dificulte cuestionar la fiabilidad de las alteraciones, tanto Mutect2 como HaplotypeCaller son sin duda herramientas potentes para la detección de variantes que deben incluirse en el flujo de trabajo.

Las dos columnas siguientes son las relacionadas con el strand bias. Por lo general, estas columnas hacen referencia a REFBIAS y ALTBIAS, las cuales muestran para cada hebra (forward:reverse) el número de lecturas que contienen la base de referencia y la variante respectivamente. No obstante, de nuevo son los dos variant callers procedentes de GATK los que proporcionan información diferente al resto.

Por un lado, Mutect2 devuelve en la primera columna un puntaje de calidad asociado a este fenómeno y en la segunda un valor booleano que sentencia la existencia o no del mismo. Por otro lado, HaplotypeCaller informa del p-valor obtenido en la prueba exacta de Fisher para tratar de detectar la aparición de strand bias y en segundo lugar ofrece la cifra numérica resultante de la prueba de razón de probabilidades (Symmetric Odds Ratio test), la cual es otro método para tratar de detectar un sesgo de hebra en los datos.

En el ejemplo, se observa como el número de lecturas que han detectado la alteración es muy parecido para ambas hebras, lo cual ha conducido a que Mutect2 niegue la existencia de strand bias devolviendo un puntaje de calidad alto, y que HaplotypeCaller obtenga valores elevados en la prueba de Fisher y bajos en la de razón de probabilidades.

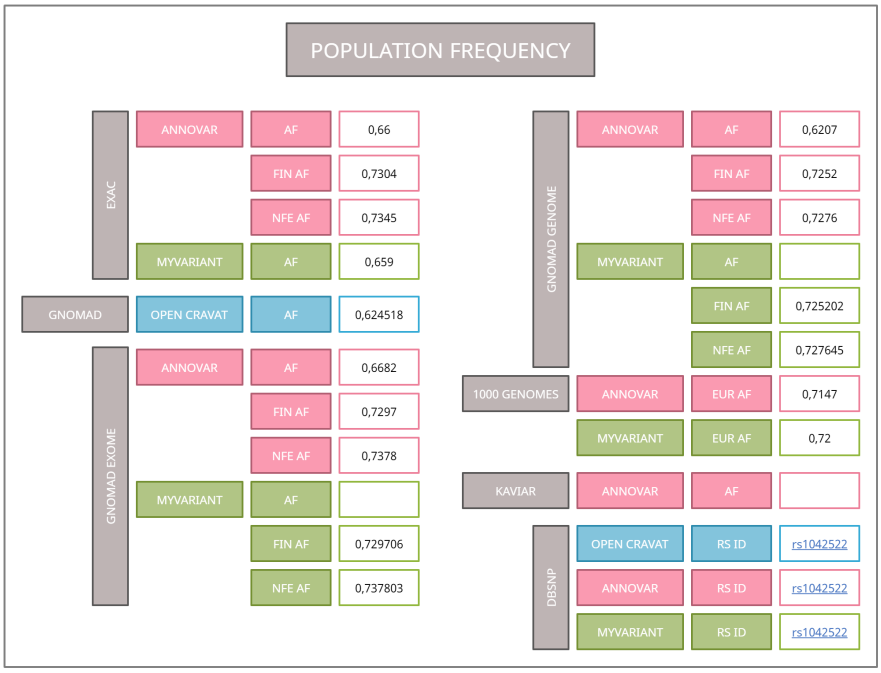


Figura 13: Segunda parte del ejemplo de una variante contenida en la tabla final del archivo XLSX. Se muestran los valores de frecuencia poblacional para las distintas bases de datos y los anotadores que han permitido el suministro de información. Se representan de color gris los nombres de las bases de datos y se le asignan los colores rosa, azul y verde a los anotadores Annovar, Open Cravat y MyVariant respectivamente.

Las siguientes celdas de la tabla se destinan a las bases de datos que ofrecen información acerca de la **frecuencia de las variantes en la población**. Tal y como se observa en la figura 13, la alteración de ejemplo parece tener una frecuencia bastante definida y unánime para todas las bases de datos contempladas. Su valor oscila entre 0,65 y 0,75, siendo por tanto una variante común en la población. Los huecos en blanco significan que la base de datos en cuestión no contempla la variante especificada y que por tanto no se le ha asignado ningún valor durante la anotación.

A pesar de que el uso de varios anotadores pueda conducir a redundancia en los resultados, no siempre es así. De hecho, en el ejemplo se observa como tanto ANNOVAR como MYVARIANT permiten la anotación de variantes con los datos de GnomAD Exome y GnomAD Genome, pero sólo ANNOVAR ha sido capaz de facilitarnos el valor de la frecuencia alélica global. Esto ocurre en muchas otras celdas de la tabla y no siempre de forma unidirec-

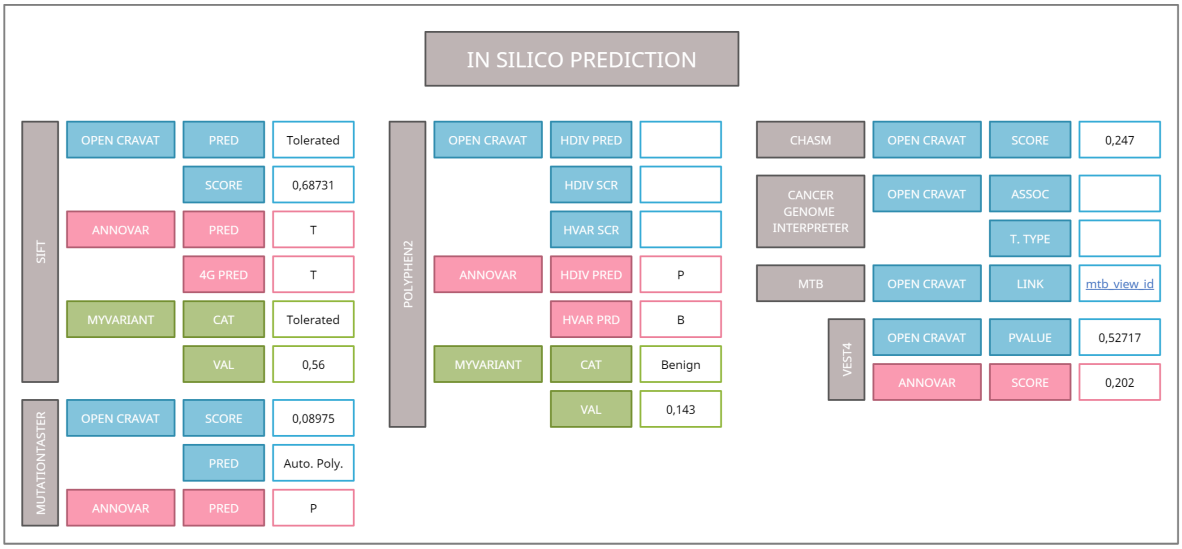


Figura 14: Tercera parte del ejemplo de una variante contenida en la tabla final del archivo XLSX. Se representan los valores categóricos y numéricos de las distintas In Silico Prediction Tools. Se mantiene la misma asignación de colores que en la figura anterior.

cional, es decir, no siempre un anotador ofrece más resultados que los demás, sino que hay datos que se encuentran en uno y no en el otro, y viceversa.

Respecto a DBSNP, el flujo de trabajo ofrece el identificador y el enlace asociado al residuo anotado. Para la variante de ejemplo expuesta en la figura, los tres anotadores han reportado el residuo rs1042522, cuyo enlace asociado es el siguiente: https://www.ncbi.nlm.nih.gov/snp/rs1042522. Esto le permite al usuario acceder de forma cómoda y directa a toda la información disponible sobre la variante en cuestión en dicha base de datos.

En las siguientes columnas del archivo se encuentran los datos ligados a las denominadas **In Silico Prediction Tools**, las cuales aportan valores numéricos y categóricos sobre la patogenicidad de las variantes. Es importante tener en cuenta que en este caso cada base de datos posee una nomenclatura y escala distinta, por lo que no es aconsejable realizar una comparación directa entre los resultados de varias columnas sin conocer la herramienta de predicción de origen.

En un intento de conocer con detalle el significado de los valores numéricos expuestos en estas celdas, en la mayoría de casos la explicación se reduce a la simple denominación

“Pathogenicity score”, sin proporcionar ningún tipo de detalle acerca de la formulación que conduce a sus resultados.

Por tanto, para poder comprender con un mínimo de coherencia los resultados obtenidos en esta parte de la tabla, se llevó a cabo un análisis exhaustivo de concordancia entre las columnas numéricas y categóricas para cada base de datos, junto con la recopilación bibliográfica pertinente en caso de ser posible:

- El score de SIFT procedente de OPEN CRAVAT no se explica en la documentación del anotador. Sin embargo, tras analizar los resultados se ha descubierto que solo se asocia a variantes dañinas cuando su valor se encuentra por encima de 0,9 y por tanto muy cercano a 1.
- El valor numérico de SIFT procedente de MYVARIANT es el que recoge la propia base de datos [142]. Su contenido posee un sentido opuesto al caso anterior, anunciando variantes dañinas cuando su valor es inferior a 0,05.
- Los valores de HDIV SCORE y HVAR SCORE de POLYPHEN2 indican la probabilidad de que una sustitución sea dañina, por lo que cuanto más cercano a 1 sea su valor más peligrosa es la alteración [143]. La única diferencia entre estos dos valores reside en sus conjuntos de datos de entrenamiento. Mientras que PolyPhen-2 HDIV utiliza alelos que codifican proteínas humanas y sus homólogos de mamíferos más cercanos, PolyPhen-2 HVAR aplica nsSNV humanos comunes como observaciones TN [144]. Tras observar detenidamente los resultados, se ha descubierto que se marcan las variantes como dañinas cuando sus valores se encuentran por encima de 0,6.
- El valor numérico de POLYPHEN2 procedente de MYVARIANT clasifica a las variantes como posibles dañinas con valores superiores a 0,5 y como probables dañinas cuando es muy cercano a 1. Es por esto, que aunque MYVARIANT no aporte información precisa de esta celda, se conoce que hace referencia al mismo valor que describe Ensembl [145].
- El score ligado a MUTATION TASTER refleja la diferencia fisicoquímica entre el aminoácido original y el mutado, siendo por tanto un valor simplemente informativo que no afecta a la predicción [146].

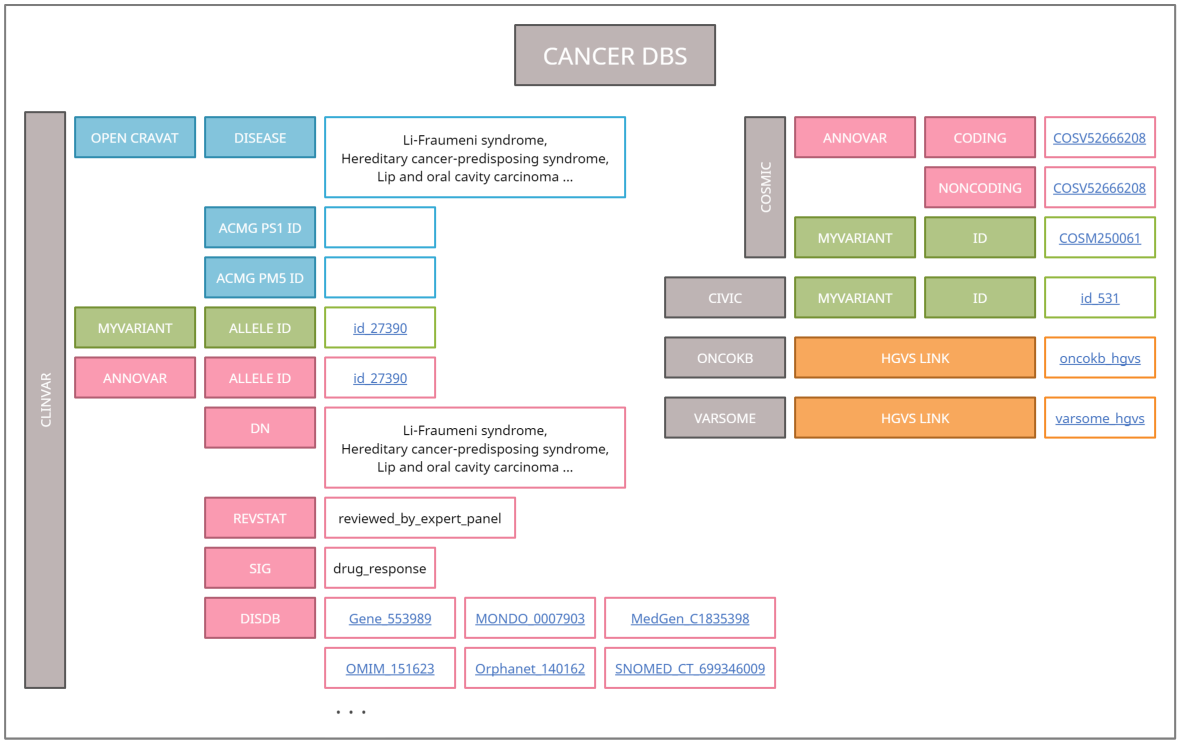


Figura 15: Cuarta parte del ejemplo de una variante contenida en la tabla final del archivo XLSX. Se muestran las anotaciones y enlaces asociadas a las distintas bases de datos especializadas en cáncer. Se mantiene la misma asignación de colores que en la figura anterior.

- Las cifras proporcionadas por CHASM se definen como la fracción de árboles que durante el algoritmo de random forest votaron a favor de que la mutación se clasificara como pasajera [147]. En la práctica, se ha observado que este valor suele resultar más alto para las variantes dañinas y más bajo para las no dañinas.

- El score devuelto por VEST4 indica el puntaje de patogenicidad de las variantes, siendo elevado cuanto más dañinas sean.

- Por último, el p-valor de VEST4 posee el mismo significado que el de SIFT, estableciendo como dañinas aquellas alteraciones con un valor inferior a 0,05.

En el ejemplo expuesto en la figura 14, se aprecia como todas las bases de datos devuelven un valor positivo para la variante. SIFT considera que es una variante tolerada por nuestro organismo, POLYPHEN2 la marca como benigna, MUTATION TASTER indica que se trata de

Celda	Enlace
ClinVar allele id	id_27390
ClinVar DisDB	Gene_553989, MONDO_0007903 MedGen_C1835398, OMIM_151623 Orphanet_140162, SNOMED_CT_699346009 etc.
COSMIC Coding and Noncoding	COSV52666208
COSMIC ID	COSM250061
CIVIC ID	id_531
OncoKB	oncokb_chr17:g.7579472G>C
VarSome	varsome_chr17:7579472:G:C

Cuadro 1: Enlaces correspondientes a los identificadores anotados en la figura anterior.

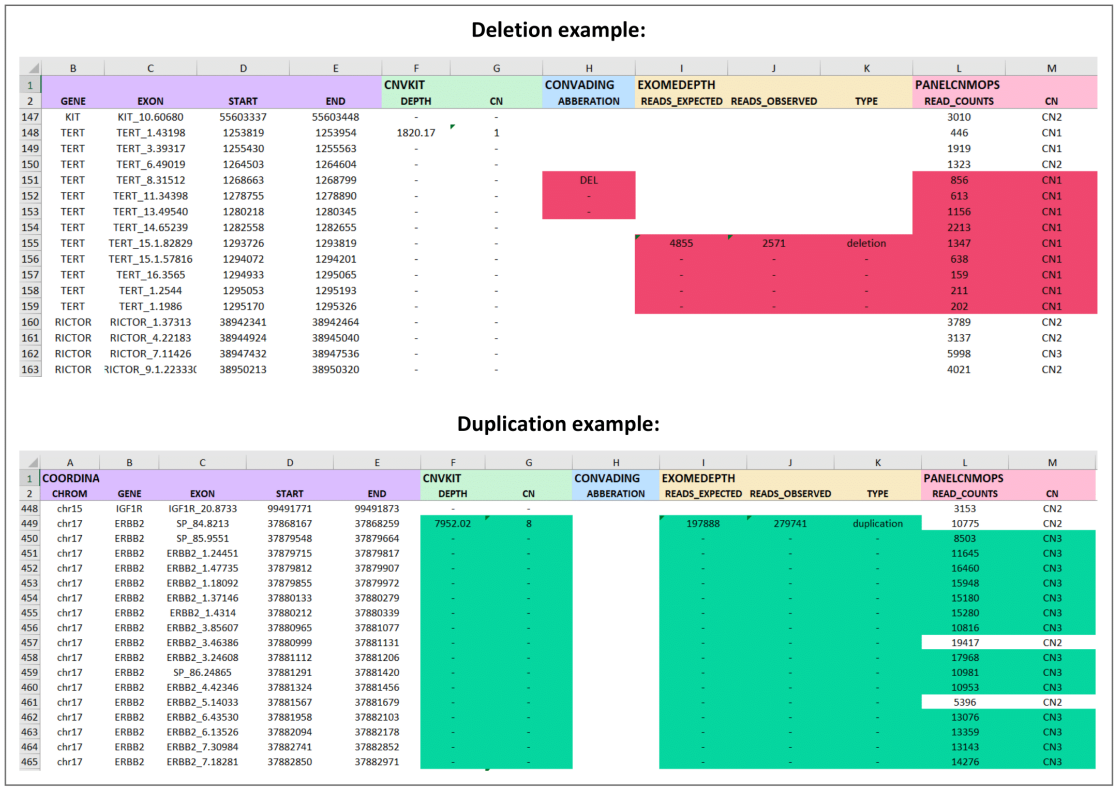
un polimorfismo automático e inofensivo y por último CHASM y VEST4 proporcionan valores de patogenicidad bajos con un p-valor elevado que descarta su peligrosidad.

Respecto a MTB (Molecular Tummor Board), el anotador OPEN CRAVAT devuelve un enlace que permite al usuario redirigirse a una página con información adicional acerca de la repercusión de la variante en la proteína afectada. En el ejemplo, el enlace es el siguiente: https://run.opencravat.org/webapps/moleculartumorboard/index.html?chrom=chr17&pos=7676154&ref_base=G&alt_base=C

Por último, están las **bases de datos especializadas en cáncer**. De ellas, lo que más interesa es extraer los identificadores asociados a las variantes con el fin de poder construir enlaces que faciliten la tarea del usuario. Un ejemplo de la cantidad de recursos que se ofrecen para cada variante, se puede ver en la figura 15, donde se ha tomado la variante usada hasta el momento. Los enlaces reportados son los expuestos en la tabla 1.

3.3.2. CNAs

Esta tabla, trata de aportar información acerca de regiones que contengan un numero de copias significativamente superior o inferior al habitual. Para ello, se le asignó una **fila a cada exón** y se adaptó la salida de todas las herramientas para poder mostrarlas correctamente en diversas columnas.



Deletion example:

	B	C	D	E	F	G	H	I	J	K	L	M
1					CNVKIT		CONVADING	EXOMEDEPTH			PANELCNMOPS	
2	GENE	EXON	START	END	DEPTH	CN	ABBERATION	READS_EXPECTED	READS_OBSERVED	TYPE	READ_COUNTS	CN
147	KIT	KIT_10.60680	55603337	55603448	-	-					3010	CN2
148	TERT	TERT_1.43198	1253819	1253954	1820.17	1					446	CN1
149	TERT	TERT_3.39317	1255430	1255563	-	-					1919	CN1
150	TERT	TERT_6.49019	1264503	1264604	-	-					1323	CN2
151	TERT	TERT_8.31512	1268663	1268799	-	-	DEL				856	CN1
152	TERT	TERT_11.34398	1278755	1278890	-	-	-				613	CN1
153	TERT	TERT_13.49540	1280218	1280345	-	-	-				1156	CN1
154	TERT	TERT_14.65239	1282558	1282655	-	-					2213	CN1
155	TERT	TERT_15.1.82829	1293726	1293819	-	-		4855	2571	deletion	1347	CN1
156	TERT	TERT_15.1.57816	1294072	1294201	-	-		-	-	-	638	CN1
157	TERT	TERT_16.3565	1294933	1295065	-	-		-	-	-	159	CN1
158	TERT	TERT_1.2544	1295053	1295193	-	-		-	-	-	211	CN1
159	TERT	TERT_1.1986	1295170	1295326	-	-		-	-	-	202	CN1
160	RICTOR	RICTOR_1.37313	38942341	38942464	-	-					3789	CN2
161	RICTOR	RICTOR_4.22183	38944924	38945040	-	-					3137	CN2
162	RICTOR	RICTOR_7.11426	38947432	38947536	-	-					5998	CN3
163	RICTOR	RICTOR_9.1.22333(	38950213	38950320	-	-					4021	CN2

Duplication example:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	COORDINA					CNVKIT		CONVADING	EXOMEDEPTH			PANELCNMOPS	
2	CHROM	GENE	EXON	START	END	DEPTH	CN	ABBERATION	READS_EXPECTED	READS_OBSERVED	TYPE	READ_COUNTS	CN
448	chr15	IGF1R	IGF1R_20.8733	99491771	99491873	-	-					3153	CN2
449	chr17	ERBB2	SP_84.8213	37868167	37868259	7952.02	8		197888	279741	duplication	10775	CN2
450	chr17	ERBB2	SP_85.9551	37879548	37879664	-	-		-	-	-	8503	CN3
451	chr17	ERBB2	ERBB2_1.24451	37879715	37879817	-	-		-	-	-	11645	CN3
452	chr17	ERBB2	ERBB2_1.47735	37879812	37879907	-	-		-	-	-	16460	CN3
453	chr17	ERBB2	ERBB2_1.18092	37879855	37879972	-	-		-	-	-	15948	CN3
454	chr17	ERBB2	ERBB2_1.37146	37880133	37880279	-	-		-	-	-	15180	CN3
455	chr17	ERBB2	ERBB2_1.4314	37880212	37880339	-	-		-	-	-	15280	CN3
456	chr17	ERBB2	ERBB2_3.85607	37880965	37881077	-	-		-	-	-	10816	CN3
457	chr17	ERBB2	ERBB2_3.46386	37880999	37881131	-	-		-	-	-	19417	CN2
458	chr17	ERBB2	ERBB2_3.24608	37881112	37881206	-	-		-	-	-	17968	CN3
459	chr17	ERBB2	SP_86.24865	37881291	37881420	-	-		-	-	-	10981	CN3
460	chr17	ERBB2	ERBB2_4.42346	37881324	37881456	-	-		-	-	-	10953	CN3
461	chr17	ERBB2	ERBB2_5.14033	37881567	37881679	-	-		-	-	-	5396	CN2
462	chr17	ERBB2	ERBB2_6.43530	37881958	37882103	-	-		-	-	-	13076	CN3
463	chr17	ERBB2	ERBB2_6.13526	37882094	37882178	-	-		-	-	-	13359	CN3
464	chr17	ERBB2	ERBB2_7.30984	37882741	37882852	-	-		-	-	-	13143	CN3
465	chr17	ERBB2	ERBB2_7.18281	37882850	37882971	-	-		-	-	-	14276	CN3

Figura 16: Ejemplo de una duplicación y una deleción observadas durante la ejecución de testeo y reportadas en el archivo XLSX final tras la detección de CNAs. Se muestran de rojo las regiones clasificadas con pérdidas y en verde las identificadas con ganancias.

En la figura 16 se muestran dos capturas correspondientes a un ejemplo de deleción y otro de duplicación. Como se puede ver, para ambos casos la detección es realizada por múltiples herramientas, lo cual aporta cierta fiabilidad a la alteración reportada. Al igual que ocurre con los variant callers, cada detector tiene su propio algoritmo de búsqueda, por lo que es habitual encontrar casos expuestos por solo una herramienta al igual que existían variantes identificadas por un solo llamador. Este fenómeno se aprecia en el artículo publicado por Lanling Zhao et al. [99] donde se muestran diversos diagramas de Venn que aunque muestren cierta concordancia entre los detectores se pueden apreciar alteraciones identificadas por una sola herramienta.

No obstante, tras el análisis de los resultados se ha comprobado que las herramientas suelen detectar de forma conjunta casos con valores significativos, es decir, alteraciones con un

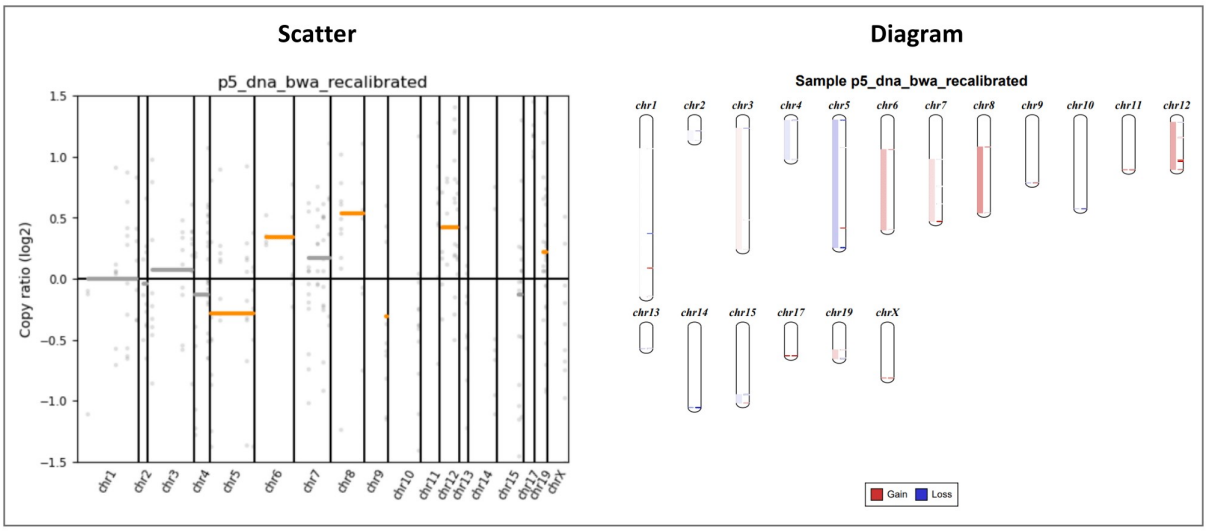


Figura 17: Se muestra un ejemplo de las dos gráficas generadas por CnvKit donde se representan el número de copias a lo largo del genoma clasificadas por cromosomas. En la imagen de la izquierda se utilizan puntos para representar el número de copias de cada exón y las rayas para indicar el promedio de cada cromosoma, destacando los más significativos mediante el color naranja. En la imagen de la derecha se realiza una representación espacial de cada cromosoma y se le asigna el rojo a las regiones con pérdidas y el azul a las zonas con ganancias, indicando una aproximación de su significancia a través de la intensidad de los colores.

claro exceso en el número de copias o una clara escasez de las mismas. Esto, permite al usuario distinguir con comodidad aquellos casos que son de alta fiabilidad y a su vez de gran trascendencia.

Respecto a los valores numéricos de profundidad de lectura, resulta complicado comparar entre distintas herramientas debido a que no todas ofrecen dicho dato y las que lo hacen cada una detecta **regiones de distinto tamaño**. Además, hay que distinguir que CnvKit devuelve valores de profundidad media, mientras que ExomeDepth y Panelcn.Mops aportan datos de profundidad total.

Sin embargo, si se hacen cálculos puede demostrarse que hay cierta concordancia entre las herramientas. Si se presta atención al caso de la deleción, ExomeDepth afirma haber encontrado un total de 2571 lecturas en su región. Esta cifra es bastante similar a la que se obtiene sumando el número de lecturas proporcionado por Panelcn.Mops para los exones contenidos en dicha región, esto es, 1347 + 638 + 159 + 211 + 202 = 2557.

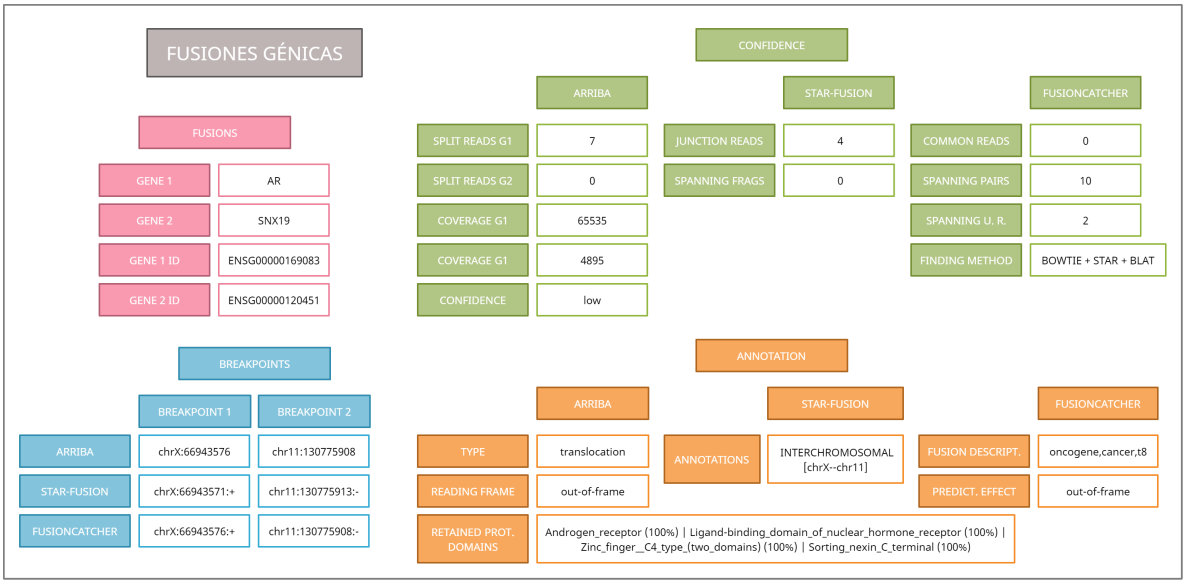


Figura 18: Ejemplo de una fusión génica detectada por todas las herramientas y reportada en el archivo XLSX final. De color rosa se muestra la información asociada a los genes implicados, de color azul los puntos de ruptura, de color verde los valores asociados a la confianza de la detección y de color naranja la anotación de la fusión génica por cada herramienta.

Además, en caso de ejecutar la herramienta CnvKit, se dispondrá en la carpeta correspondiente dos gráficas adicionales que facilitan la comprensión de los resultados devueltos por esta misma herramienta. Un ejemplo de estas gráficas se contempla en la figura 17, donde se puede apreciar como ambas tratan de representar el número de copias asociado a cada cromosoma pero utilizando ideas visuales distintas.

3.3.3. Reordenamientos genéticos

Por último, la tabla referente a fusiones génicas muestra una lista de fusiones candidatas con sus respectivos **puntos de ruptura**, **confianza** y **anotación** para cada detector. A diferencia de los casos anteriores, puesto que este tipo de alteración genética es más inusual en las muestras, es bastante habitual que alguna herramienta no sea capaz de detectar nada o que se reporten fusiones de muy baja cobertura.

Además, dado que tal y como se expuso en la metodología, ninguna de las tres herramientas se especializa del todo en la modalidad de secuenciación del laboratorio, sus resultados suelen

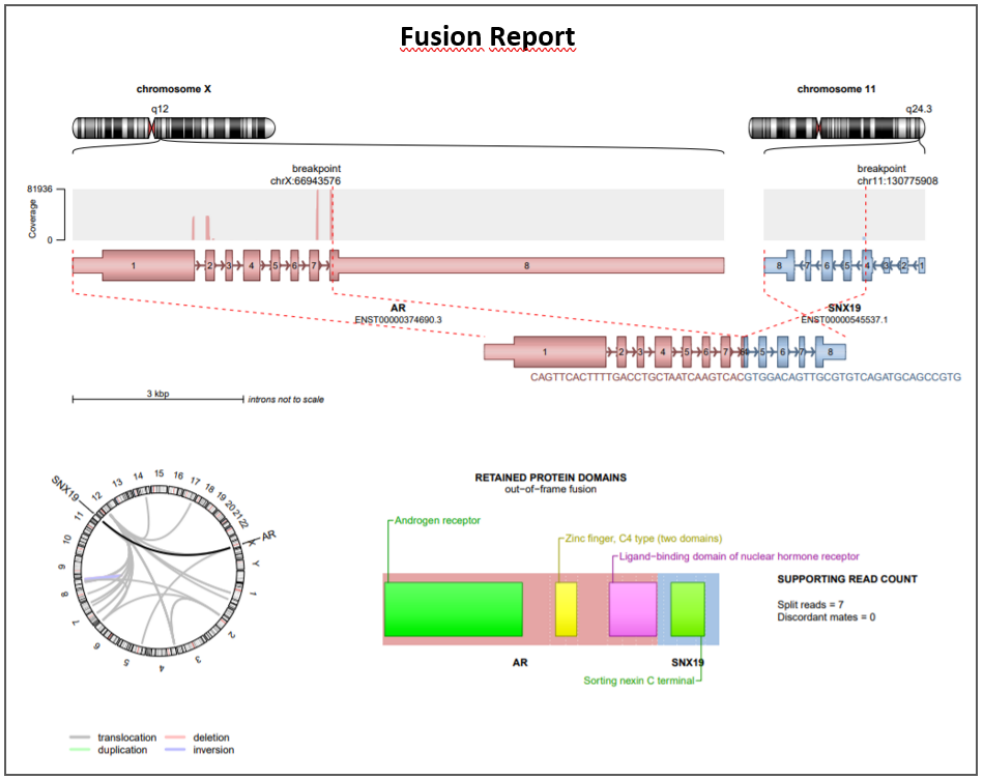


Figura 19: Diapositiva generada por Arriba para la fusión génica expuesta en la anterior figura.

ser poco coincidentes y bastante sensibles al alineamiento realizado.

Para mostrar un ejemplo de una fusión génica detectada, se han unido sus 27 celdas de información en una única figura con el fin de facilitar su visualización. Esta es la figura 18, la cual expone una fusión génica de muy baja fiabilidad pero detectada por todas las herramientas.

En ella, se puede apreciar la unión del gen AR propio del cromosoma X con el gen SNX19 perteneciente al cromosoma 11. Si se presta atención a los puntos de ruptura, se observa como los tres detectores coinciden a la hora de realizar su posicionamiento, estableciendo el punto del primer gen en torno a la posición 66943576 del cromosoma X y el punto del segundo gen cercano a la posición 130775908 del cromosoma 11.

Esta coherencia entre las tres herramientas, también se extiende a la sección de confianza donde todas ellas exponen un bajo número de lecturas que respaldan la aparición de esta fusión. De hecho, se observa como Arriba la clasifica directamente de confianza tipo “low”.

Respecto a la anotación, la información reportada por los distintos detectores indica que se trata de una translocación fuera del marco de lectura cuya repercusión se encuentra ligada al cáncer.

Además, en caso de ejecutar Arriba, se dispondrá en su carpeta correspondiente un archivo PDF con información de las distintas fusiones génicas detectadas por esta misma herramienta. Si se coge el mismo ejemplo que antes, Arriba muestra lo ilustrado en la figura 19. En ella, se permite tener una visión más espacial de la fusión génica, así como conocer la secuencia exacta que ha conducido a su detección junto con el origen de los datos de anotación.

3.4. Comparación con los resultados de Ion Reporter

Una vez comprendida la salida del script, resulta fundamental llevar a cabo una comparación de resultados con el software actual instalado en el laboratorio, es decir, Ion Reporter. Para ello, se realizarán diversos estudios que traten de procesar las salidas de cada pipeline con el fin de equiparar sus formatos y poder realizar una comparación justa y adecuada. Al igual que siempre, se dividirá el análisis en las tres secciones habituales: SNVs + INDELs, CNAs y Reordenamientos Genéticos.

3.4.1. SNVs + INDELs

Para realizar esta comparación y situar los resultados de ambos flujos en igualdad de condiciones, se debe realizar un procesamiento previo individual e independiente para cada salida.

Por un lado, se debe recordar que en la ejecución del script el filtrado de las variantes se realiza ya en la fase final durante la construcción del archivo XLSX, por lo que no se posee un VCF con las variantes definitivas. Y por otro lado, aunque la salida de Ion Reporter sí devuelva dicho fichero, no se encuentra normalizado y contiene más de una variante por línea. Por tanto, la comparación más adecuada consiste en el procesamiento por separado de ambos archivos en R para obtener las columnas de CHROM, POS, REF y ALT correspondientes a cada salida y así poder compararlas entre ellas.

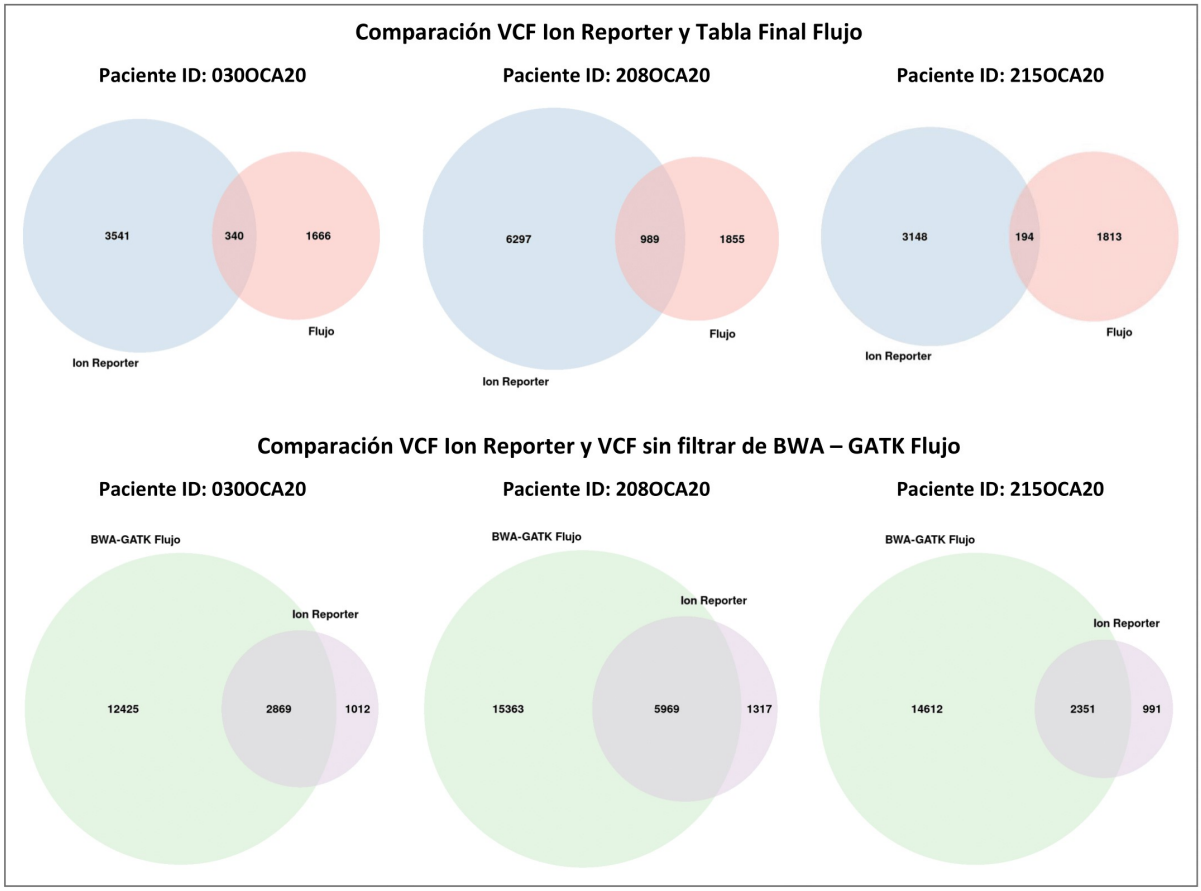


Figura 20: Comparación de SNVs e INDELs entre el flujo de trabajo e Ion Reporter para tres pacientes concretos de la ejecución de testeo. En la primera línea tenemos la comparación entre la tabla final del flujo de trabajo y el VCF generado por Ion Reporter con las variantes filtradas, y en la segunda los diagramas de Venn asociados a la comparación entre las variantes de Ion Reporter y las generadas por las combinaciones de BWA-GATK del flujo de trabajo antes de realizar su filtrado.

Esto se realiza con el script **comparison_snv+indels.R**, el cual una vez que obtiene las cuatro columnas para cada pipeline, las une en una única tabla para identificar el número de duplicados y con ello el número de variantes comunes en ambas salidas.

Tras esto, se hace uso del paquete **VennDiagram** de R para graficar los diagramas de Venn pertinentes. Estos diagramas pueden observarse en la primera fila de la figura 20, donde se compara la salida de Ion Reporter y el flujo de trabajo para tres pacientes concretos de la ejecución de testeo. En ellos, se aprecia como el número de variantes que ambas salidas tienen en común es bastante pequeño, representando una media del 15 % de la salida del script ejecutado en Picasso.

ID paciente	Ion Reporter	Tabla Final	Inter. Total	Inter. BWA-GATK
030OCA20	3881	2006	340	2869
068OCA21	2473	2408	110	2087
070OCA21	2459	2289	108	2126
199OCA20	2452	2151	107	2064
208OCA20	7286	2844	989	5969
215OCA20	3342	2007	194	2351
220OCA20	3075	2128	168	2160

Cuadro 2: Datos de la comparación de SNVs e INDELs entre el flujo de trabajo e Ion Reporter parra todos los pacientes de la ejecución de testeo.

Aunque esto pueda parecer alarmante en primera instancia, se realizó un estudio más exhaustivo que tratase de explicar este resultado. Tras hablar con varios informáticos que han tratado este tema, se extrajo la información de que el flujo de trabajo implementado en Ion Reporter es una modificación personalizada del uso conjunto del mapeador BWA y los llamadores de variantes de GATK.

Por tanto, dado que el flujo de este trabajo contempla dichas herramientas, resulta pertinente realizar la misma comparación que antes pero sustituyendo las variantes definitivas por la unión de las contenidas en los VCF resultantes de las combinaciones BWA-GATK_Mutect2 y BWA-GATK_HaplotypeCaller ejecutadas antes de las fases de intersección, unión y filtrado.

Como se puede ver en la segunda fila de la figura, el resultado es completamente distinto al anterior. Prácticamente la totalidad de las variantes devueltas por Ion Reporter se encuentran en alguno de los dos archivos VCF generados por las combinaciones mencionadas. La explicación de que estas variantes no sean finalmente reportadas por el flujo de trabajo es que la mayor parte de ellas no sobrevive a la intersección con otros mapeadores o al proceso de unión y filtrado posterior basado en umbrales de media de frecuencia alélica y profundidad de lectura, así como su posicionamiento en regiones codificantes o clasificación en sequence ontology.

Además, las pocas variantes que los diagramas representan como no contenidas en las combinaciones BWA-GATK del flujo, no elimina la posibilidad de que sí aparezcan en alguna del resto de combinaciones de otras herramientas, y que al igual que las anteriores hayan

podido ser eliminadas mediante intersecciones y filtrados.

En resumen, las variantes de Ion Reporter sí que son contempladas por el flujo de trabajo de este proyecto, solo que al utilizar un mayor número de herramientas se obtienen un mayor volumen de variantes que permite ser más estricto durante el filtrado de las mismas. Por tanto, el flujo no solo solapa el campo de búsqueda de Ion Reporter sino que contiene muchas más combinaciones que exploran variantes a través de otros algoritmos igualmente fiables que no son contemplados por el software actual del laboratorio.

Dado que este estudio se realizó para más pacientes, en la tabla 2 se muestran sus resultados detallados con el fin de verificar el fenómeno descrito para otros casos.

3.4.2. CNAs

En este caso, dado que Ion Reporter devuelve para cada paciente un archivo tabulado con el número de copias detectadas en cada exón, la mejor comparación posible es añadir sus datos en una columna adicional de la tabla contenida en el archivo XLSX. Para ello, se hará uso del script de R **comparison_cnvs.R** para analizar visualmente la concordancia entre resultados a través de la tabla construida.

Durante la comprobación manual de resultados, se ha podido apreciar cierta relación entre la salida del pipeline diseñado y la de Ion Reporter. Algunos ejemplos que verifican la concordancia detectada se exponen en la figura 21, donde la última columna de cada captura muestra los datos proporcionados por el software del laboratorio. A pesar de que los valores numéricos no coincidan de forma exacta sí que parecen detectar las mismas regiones alteradas, tanto en casos de deleciones (parte izquierda) como de duplicaciones (parte derecha).

No obstante, al igual que en la sección anterior existen algunas excepciones en las que los resultados parecen no estar de acuerdo. Se han encontrado casos pertenecientes a las dos situaciones siguientes:

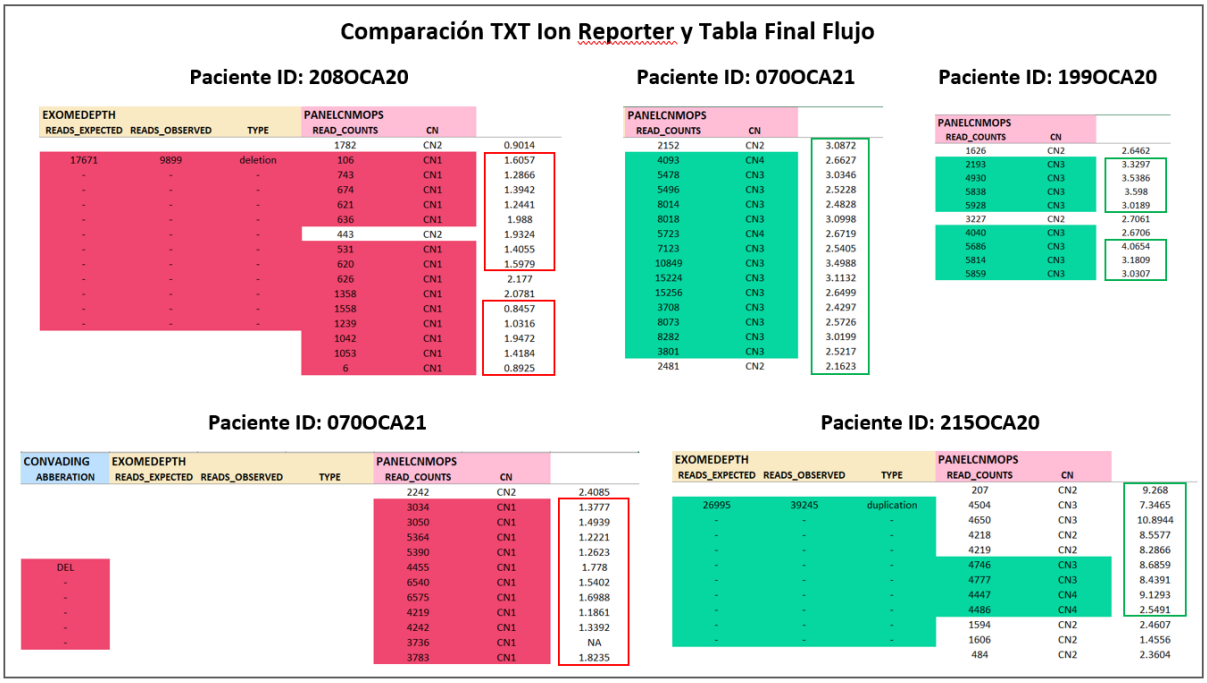


Figura 21: Comparación de CNAs entre la salida del flujo de trabajo e Ion Reporter. En la imagen se muestran 5 casos de concordancia entre ambos pipelines para distintos pacientes. A la izquierda se presentan dos ejemplos de deleciones y a la derecha tres de duplicaciones.

- Alteraciones detectadas por alguno de los dos flujos pero no detectado por el otro.
- Número de copias contradictorias que en una salida se presentan como duplicaciones y en la otra como deleciones.

Cabe mencionar que los casos de la primera situación no suelen ser alteraciones significativas, presentando usualmente un número de copias cercano a 2, y por otro lado, los casos de la segunda situación son bastante escasos y suelen localizarse en regiones donde para todos los pacientes las propias herramientas del flujo también pierden cierta concordancia.

Mientras que la primera circunstancia se relaciona con la sensibilidad de las herramientas, la segunda se asocia a la referencia construida. El hecho de que sea una zona conflictiva para todos los pacientes quiere decir que la referencia diseñada a partir de sus datos no presenta unos niveles adecuados para ser utilizados como un punto de partida en dicha región.

Un ejemplo de este fenómeno podría ser que la mayoría de pacientes mostrasen un número de copias elevado en un mismo exón. Esto provocaría que el flujo de trabajo al construir

la referencia asumiera en dicha zona esos valores elevados como normales y que cualquier paciente con un número de copias verdaderamente normal se le asignase una deleción cuando no la tiene. La forma en la que cada herramienta construya la referencia determinará por tanto los resultados en estos casos.

Respecto a Ion Reporter, dado que se desconoce por completo su procedimiento en la detección de este tipo de alteraciones, no se sabe su sensibilidad ante estos casos o incluso si se apoya de una referencia previamente establecida por la propia casa comercial.

En resumen, se considera que hay una concordancia adecuada entre ambos flujos de trabajo a excepción de casos esporádicos dependientes de las propias muestras y de la sensibilidad de los detectores.

3.4.3. Reordenamientos Genéticos

Tal y como se ha mencionado en apartados anteriores, este tipo de alteración genética es poco frecuente en las muestras. De hecho, de todos los datos proporcionados por Ion Reporter solo ha habido una fusión génica que ha superado su filtro de calidad. Esta fusión corresponde al paciente 070OCA21 y representa la unión de los genes MIR143HG (chr5:148786641) y NOTCH1.M1N27 (chr9:139397782). Dicha alteración también es detectada por el flujo de trabajo para este paciente, sin embargo, esta coincidencia no es suficiente para analizar la concordancia entre ambos pipelines.

Para poder realizar una comparación coherente, se debe analizar previamente la salida de cada flujo de trabajo. Por un lado, el resultado de Ion Reporter se proporciona en un fichero TSV que muestra todas las alteraciones detectadas junto con su clasificación respecto al filtro aplicado (NOCALL o PASS). Y por otro lado, el script de este proyecto proporciona en el fichero XLSX una lista de fusiones génicas con ciertos valores de confianza asociados a la herramienta que haya realizado la detección.

Aunque ambos resultados ofrezcan información acerca de la fiabilidad de estas alteraciones, en los dos casos se presenta la lista completa de fusiones candidatas. Por tanto, dado

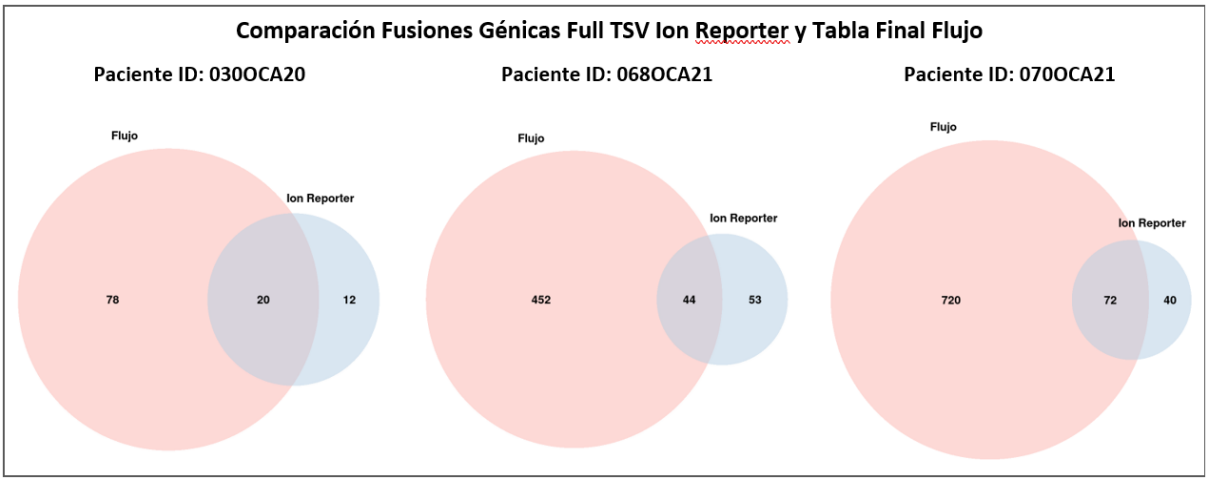


Figura 22: Comparación de las fusiones génicas candidatas del flujo de trabajo con las de Ion Reporter. Se muestran los diagramas de Venn para tres pacientes concretos generados tras la ejecución del script de R pertinente.

que la comparación de fusiones fiables ya ha sido demostrada como insuficiente, se tratará de comparar la lista completa proporcionada por ambos flujos de trabajo.

Para ello, se ha diseñado el script **comparison_fusions.R**, el cual posee una implementación parecida al del apartado de SNVs + INDELs solo que en este caso se ha utilizado un sistema de comparación un poco más elaborado. Por lo observado en los resultados de la ejecución en Picasso, los puntos de corte de una misma fusión génica pueden variar unas cuantas bases dependiendo del detector que la identifique. Es por esto, que el script diseñado permite un margen de 10 bases para considerar a dos fusiones génicas como iguales.

En la figura 22 se muestran los diagramas de Venn para tres pacientes concretos de la ejecución de testeo. Como puede apreciarse, la concordancia entre resultados no es del todo favorable, cubriéndose en algunos casos solo la mitad de las fusiones candidatas de Ion Reporter. No obstante, se debe recordar que todas las fusiones comparadas son poco fiables y que por tanto algunas de ellas pueden proceder de lecturas mal alineadas que realmente no deben aparecer en los resultados del otro flujo de trabajo.

De igual forma, el hecho de existir un número significativo de fusiones comunes, permite sospechar que hacen referencia a aquellas cuya detección resulta más evidente y que por tanto pertenecen a casos que poseen un mayor número de lecturas de respaldo y confianza.

ID paciente	Ion Reporter	Tabla Final	Intersección
030OCA20	32	98	20
068OCA21	97	496	44
070OCA21	112	792	72
199OCA20	12	45	9
208OCA20	41	320	22
215OCA20	92	544	57
220OCA20	103	681	76

Cuadro 3: Datos de la comparación de fusiones génicas entre el flujo de trabajo e Ion Reporter parra todos los pacientes de la ejecución de testeo.

Como era de esperar, el número de fusiones candidatas del script de este trabajo es mucho mayor que el de Ion Reporter. La explicación reside en que el flujo de trabajo construido integra un total de tres herramientas distintas para llevar a cabo la detección, lo cual devuelve un volumen de resultados bastante mayor que el de una sola.

Para poder visualizar esta comparación en otros pacientes, se han expuesto los resultados en la tabla 3, donde se aprecian cifras similares a las representadas en los diagramas anteriores.

3.5. Tiempo de ejecución

Tras exponer el contenido de los resultados proporcionados por el script, resulta fundamental comentar acerca del tiempo de ejecución requerido y por tanto ver si es una solución factible para su uso clínico en el laboratorio. Teniendo en cuenta que el flujo de trabajo va a resultar siempre ejecutado bajo las mismas peticiones de número de núcleos (32) y memoria RAM (700 GB), el tiempo de ejecución depende de dos factores principales: el **número de muestras** y la **cantidad de herramientas** seleccionadas.

3.5.1. Mapeadores

El flujo de trabajo ofrece una amplia **diversidad de mapeadores** en su implementación. Sin embargo, ejecutar todos ellos simultáneamente es perjudicial para los resultados a la vez que incrementa de forma innecesaria el tiempo de ejecución. Por ello, se debe realizar una

Mapeador	Tiempo generación índice	Tiempo alineamiento
BWA	00:57:27	00:01:42
Bowtie2	00:36:43	00:02:00
Gmap	01:01:59	03:05:01
Subread	00:15:24	00:05:24
Hisat2	00:18:50	00:00:48
Novoalign	00:01:23	10:12:09
Gem3	00:11:07	00:00:43
Kart	00:58:04	00:00:56

Cuadro 4: Tiempo de ejecución para cada alineador

selección predeterminada de este tipo de herramientas basándose en lo visto en los informes de calidad de Qualimap y en el tiempo de ejecución abordado en este apartado.

Tal y como se muestra en la tabla 4, los tiempos de ejecución para la generación de índices asociados a hg19 oscilan entre pocos minutos y 1 hora. El mapeador que presenta una mayor rapidez en esta tarea es Novoalign mientras que los más lentos son Kart, BWA y Gmap.

A pesar de que la generación del índice es solo necesaria durante la primera ejecución del flujo, el tiempo que dedica cada mapeador a esta tarea suele ser equivalente a lo bien que cada uno de ellos se facilita el posterior alineamiento de las lecturas.

Si se presta atención a los tiempos de la última columna, se aprecia como Novoalign es un claro ejemplo de lo dicho anteriormente. Es decir, consigue construir el índice del genoma en muy poco tiempo, pero después tarda mucho más que los demás en llevar a cabo el alineamiento de las lecturas. Este comportamiento tiene antecedentes reportados en artículos como el escrito por Sophie Schbath et al. [148] donde en su tabla número 3 se puede apreciar que NovoAlign obtiene un tiempo de ejecución muy pequeño para la construcción del índice y otro extremadamente grande para el alineamiento de las lecturas.

Dado que en este caso la referencia es fija y la construcción de su índice solo se realiza una vez, no es relevante considerar el tiempo que tardan las herramientas en llevarlo a cabo para realizar una selección. Es por esto, que la elección del subconjunto se basará principalmente en la última columna de la tabla.

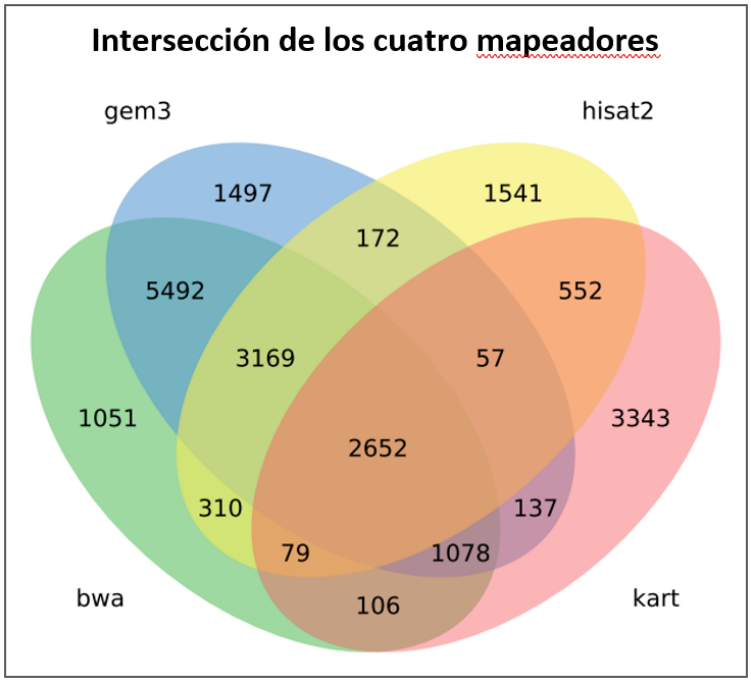


Figura 23: Diagrama de Venn correspondiente a la intersección de las variantes procedentes de los cuatro mapeadores finalistas para una muestra concreta de la ejecución de testeo.

Si se recuerda lo mencionado en los informes de Qualimap, los mapeadores que mejores resultados obtienen son Novoalign, BWA, Hisat2, Gem3 y Kart. De todos ellos, aplicando la nueva información adquirida respecto a los tiempos de ejecución, se debe eliminar Novoalign ya que a pesar de su buena calidad se trata del **mapeador más lento** con diferencia.

De los cuatro restantes, tal y como se observa en la figura 23, se comprobó que Kart era el que **menos variantes coincidentes** reportaba respecto a los otros tres mapeadores. Por ello, también fue descartado de la lista definitiva, dejando la combinación de Bwa, Gem3 y Hisat2 como el subgrupo de alineadores óptimo para las muestras del laboratorio.

3.5.2. Tiempo de ejecución por paciente

Respecto al resto de herramientas, puesto que la aplicación de todas ellas siempre es beneficioso para la selección de variantes por parte del usuario, se ha decidido **mantenerlas todas** por defecto en el fichero de configuración.

Si reducimos el flujo a un único paciente con su respectivas muestras de DNA y RNA, el tiempo de ejecución viene definido en su correspondiente archivo **runtime.txt**. En la tabla 5 se muestra lo obtenido para el paciente 030OCA20 de la ejecución de testeo.

Tarea	**Herramienta**	**Hebras**	**Tiempo**
Procesamiento de lecturas			
- Informes lecturas iniciales	FastQC	multithread	00:00:19
- Cortar adaptadores	Cutadapt	multithread	00:00:05
- Trimming, Long Min, Indet, Cont GC	PrinSeq	multithread	00:00:16
- Informes lecturas procesadas	FastQC	multithread	00:00:19
- Informes comparativos	MultiQC	single thread	00:00:08
Mapeo			
- Alineamiento de lecturas DNA	BWA	multithread	00:00:17
	Hisat2	multithread	00:00:20
	Gem3	multithread	00:00:09
Procesamiento de archivos bam DNA			
- Compresión SAM ->BAM	Samtools view	multithread	00:00:06
- Grupos de lectura	GATK	single thread	00:00:33
- Ordenar lecturas	Samtools sort	multithread	00:00:07
- Recalibración de calidad de bases	GATK	parallel	00:02:46
- Informes bam DNA	Qualimap	single thread	00:02:44
SNVs + INDELs			
- Llamada de variantes SNVs + INDELs	VarDict	multithread	00:01:29
	Mutect2		
	HaplotypeCaller		
	VarScan	parallel	00:20:22
	Freebayes		
	LoFreq		
- Diagramas de Venn (intersección)	Vcftoolz	single thread	00:00:17
- Intersección para cada variant caller	Vcf-isec	single thread	00:00:05

Tarea	Herramienta	Hebras	Tiempo
- Diagrama de Venn (unión)	Vcftoolz	single thread	00:00:05
- Unión de intersecciones	Bcftools	multithread	00:00:01
Anotación de variantes			
- Anotación (SNVs + INDELs)	Annovar	single thread	00:03:08
	Open Cravat	single thread	00:03:20
- Crear xls consenso	R Rscript	partial multithread	00:09:52
CNAs			
- Detección de CNAs	CnvKit	single thread	00:00:13/7
	CoNVaDING	single thread	00:04:47/7
	ExomeDepth	single thread	00:04:10/7 + 00:00:10
	Panelcn.Mops	single thread	00:00:53/7 + 00:00:14
- Unir CNAs	Rscript	single thread	00:00:02
Fusiones Génicas			
- Detección de Fusiones Génicas	Arriba	8 threads	00:08:48
	Star-Fusion	4 threads	00:02:23
	FusionCatcher	multithread	00:22:22
- Unir Fusiones Génicas	Rscript	single thread	00:00:02

Cuadro 5: Contenido del archivo runtime.txt del paciente con ID 030OCA20. En él se muestra el tiempo requerido por cada herramienta durante su ejecución.

Si se suman los tiempos de todas las operaciones, se obtiene un total de 1 hora 22 minutos y 14 segundos. Esto significa que la ejecución habitual del laboratorio formada por 16 pacientes tardaría en torno a 21 horas, es decir, menos de un día. A pesar de parecer mucho tiempo, el software de **Ion Reporter** integrado actualmente en el laboratorio tarda unos dos días, por lo que el flujo de trabajo de este proyecto es considerablemente más rápido.

Aunque la mayoría de herramientas suela presentar un tiempo de ejecución similar para

todas las muestras, en el caso de la detección de fusiones génicas se ha podido comprobar como las herramientas Arriba y FusionCatcher son claramente sensibles al tamaño del archivo FASTQ de RNA. Por tanto, la aproximación anterior no es del todo fiable y puede variar dependiendo del número de lecturas de las muestras de entrada.

Como ya se ha comentado anteriormente, el uso de tantas herramientas en un tiempo de ejecución viable, es posible gracias al acceso a la **supercomputadora Picasso**, la cual ofrece la posibilidad de ejecutar el script en un entorno de 32 cores y 700 GB de RAM. No obstante, al igual que el resto de usuarios de Picasso, el tiempo que se tarda en obtener los resultados no es el mismo que el tiempo que el script tarde en ejecutarse. Esto se debe al **sistema de colas** establecido en la supercomputadora que asigna las preferencias por orden de llegada.

Por tanto, el tiempo total para obtener los resultados es el tiempo de espera en la cola más el tiempo de ejecución. Aunque el segundo sumando pueda ser estimado por el laboratorio, el primero depende por completo del resto de usuarios y sus respectivas ejecuciones.

4

Conclusiones y Líneas Futuras

4.1. Conclusiones

El flujo de trabajo diseñado es capaz de analizar los datos procedentes de la secuenciación masiva dirigida con panel de 161 genes de Ion Torrent para identificar alteraciones moleculares en pacientes oncológicos. Este flujo ha demostrado que proporciona resultados fiables que permiten tanto la verificación de las variantes detectadas por Ion Reporter como el descubrimiento de nuevas alteraciones genéticas hasta ahora no identificadas por el software comercial. Todo ello, en un tiempo de ejecución viable que equipara o incluso mejora el requerido por la plataforma actual gracias al acceso de Picasso y una codificación centrada en la implementación de herramientas multihebra y el empleo del paralelismo.

El script diseñado ha conseguido eliminar el mencionado concepto de "caja negra" y superar la desinformación proporcionada por otros pipelines. Esto ha sido posible gracias a un diseño flexible reflejado en el fichero de configuración, la proporción de diversos documentos y gráficas que representan la calidad de los resultados, exposición detallada de los tiempos de ejecución consumidos por cada fase y la especificación de las herramientas que han originado cada uno de los datos reportados en las tablas finales.

Se le ha otorgado al flujo una salida con mucha más información para cada variante que la que se obtiene con el software actual gracias a la consulta de numerosas bases de datos a través de múltiples anotadores. De hecho, además de ganar en cantidad de información, la tarea realizada durante la asignación de enlaces, incrementa la comodidad de análisis por parte

del usuario, que a diferencia de antes, no tiene la necesidad de estar consultando manualmente los identificadores obtenidos en cada base de datos.

4.2. Líneas Futuras

Respecto a posibles modificaciones que se puedan realizar en futuras versiones de este flujo de trabajo, se proponen seis ideas cuyo objetivo es mejorar la calidad de los resultados y optimizar aun más el tiempo de ejecución.

4.2.1. Paralelismo

A pesar de haber aplicado este concepto en nuestro script, solo se ha abordado en las zonas más críticas del mismo, como por ejemplo en la unificación de resultados con **mcapply()** o en la fase asociada a la identificación de SNVs e INDELs donde la ejecución de varios llamadores de variantes se realiza bajo **GNU parallel**.

No obstante, esta práctica podría **extenderse a otras zonas** de nuestro flujo de trabajo, que a pesar de consumir menos tiempo también pueden optimizarse. Puntos potenciales dispuestos a sufrir este cambio son la anotación de variantes, la detección de CNAs o la identificación de fusiones génicas.

Además, se podría diseñar de alguna forma un **paralelismo más inteligente** que jugase con el equilibrio entre multihebra y GNU parallel de tal forma que permitiese la puesta en marcha simultanea de operaciones referentes a distintas muestras. Sin embargo, a su vez tendría que seguir optimizando la ejecución para una sola muestra y asegurarse la correcta construcción de la referencia en la fase de detección de CNAs.

Un ejemplo podría ser paralelizar la ejecución de varios pacientes, y el número de threads que queden disponibles para cada uno de ellos que fuesen utilizados por las herramientas multihebra y las demás paralelizarlas con GNU parallel dentro del propio paralelismo inicial de los pacientes.

4.2.2. Complementariedad entre mapeadores y llamadores de variantes

Desde la primera etapa del proyecto hasta la exposición de resultados, las herramientas integradas en el flujo de trabajo han sido valoradas por sus méritos propios e individuales, siendo elegidas por su prestigio en la bibliografía y mantenidas en el proyecto tras el análisis de la calidad de sus resultados mediante herramientas como FastQC o Qualimap.

El empleo de esta visión reduccionista no contempla la opción de que la concatenación de dos herramientas de menor prestigio pero mayor complementariedad entre ellas, pueda obtener mejores resultados que la unión de dos herramientas de alta calidad que hayan destacado por sus resultados individuales.

La idea que se propone en este apartado, es analizar la complementariedad existente entre nuestros alineadores y variant callers, para que en lugar de ejecutar todas las combinaciones posibles sólo se utilicen las combinaciones que hayan demostrado proporcionar los resultados más fiables. Esto lograría reducir el tiempo de ejecución a la vez que se eliminan falsos positivos causados por malas combinaciones hasta ahora no identificadas.

4.2.3. Aprendizaje computacional

En caso de poseer muestras con variantes ya conocidas y querer profundizar aún más la idea anterior, se podrían emplear estrategias propias del aprendizaje computacional para identificar de forma automática dichas combinaciones prometedoras o incluso personalizar la elección de las mismas en función de características de la muestra de entrada como la longitud de lectura o la cobertura.

No obstante, con la tecnología y capacidad actual, conllevaría una codificación bastante compleja con un tiempo de ejecución inviable para su uso clínico.

4.2.4. Mayor flexibilidad

Esta idea se centra en proporcionar una mayor flexibilidad al script, ofreciendo nuevas opciones actualmente no disponibles. Algunos ejemplos de estas opciones son:

- Permitir el uso de nuevas referencias mediante su indicación en el fichero de configuración, de tal forma que el flujo de trabajo identifique de forma automática los archivos que debe consultar durante la recalibración de la calidad de las bases o la anotación de variantes.
- Modificación del archivo BED correspondiente al panel genético, pudiendo añadir su ruta al fichero de configuración y por tanto cambiar de manera automática las regiones observadas durante la detección de alteraciones genéticas.
- Permitir la entrada de lecturas ya alineadas y comenzar la ejecución en un punto del script distinto al inicial. De hecho, se podrían mezclar ambos tipos de entrada de tal forma que el flujo detectase las terminaciones de los ficheros (.fastq o .bam) y estableciese el inicio para cada paciente en el punto de ejecución pertinente.
- Hacer posible configuraciones personalizadas para cada muestra con el fin de que se ejecuten las combinaciones de mapeadores y detectores que el usuario establezca como más adecuadas para cada una de ellas.
- Establecer una lista de bases de datos durante la anotación, de tal forma que el usuario pueda elegir cuales consultar.

4.2.5. Interfaz gráfica

Como cualquier herramienta destinada a usuarios sin conocimientos informáticos, es siempre favorable diseñar una interfaz gráfica que facilite su uso. De esta forma, el fichero de configuración podría tomar forma de formulario web y los archivos destinados a registrar el tiempo de ejecución de cada fase podrían transformarse en gráficas que informasen acerca del progreso de cada ejecución.

Además, la proporción de resultados podría integrar la conversión de los archivos XLSX a formato HTML para su posible visualización en la interfaz, e incluso introducir opciones de filtrado y búsquedas avanzadas sobre las variantes detectadas.

4.2.6. Script de actualización

De forma adicional al flujo de trabajo principal, sería conveniente el diseño de otro script destinado exclusivamente a la actualización automática de los archivos de referencia ligados a procesos como la recalibración de la calidad de las bases o la anotación de variantes.

No obstante, esta tarea resulta bastante compleja si tenemos en cuenta que no todas las bases de datos aceptan la descarga de ficheros a través del terminal sin un inicio de sesión previo como COSMIC. Además, la búsqueda de nuevas versiones puede resultar complicada ante la creciente aparición de nuevas bases de datos o unión de algunas ya existentes con la asignación de nuevos nombres y direcciones web.

Referencias bibliográficas

[1] "Picasso | res - red española de supercomputación." [Online]. Available: https://www.res.es/es/nodos-de-la-res/picasso

[2] "El laboratorio de biología molecular del cáncer de la uma, referente en técnicas moleculares de vanguardia - universidad de málaga." [Online]. Available: https://www.uma.es/sala-de-prensa/noticias/el-laboratorio-de-biologia-molecular-del-cancer-de-la-uma-referente-en-tecnicas-moleculares-de-va

[3] O. Tan, R. Shrestha, M. Cunich, and D. J. Schofield, "Application of next-generation sequencing to improve cancer management: A review of the clinical effectiveness and cost-effectiveness," pp. 533–544, 3 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1111/cge.13199https://onlinelibrary.wiley.com/doi/abs/10.1111/cge.13199https://onlinelibrary.wiley.com/doi/10.1111/cge.13199

[4] P. Hu, F. Qiao, Y. Wang *et al.*, "Clinical application of targeted next-generation sequencing in fetuses with congenital heart defect," *Ultrasound in Obstetrics and Gynecology*, vol. 52, pp. 205–211, 8 2018. [Online]. Available: http://www.bioinformatics.org/wiki/CHDWiki

[5] J. Rexach, H. Lee, J. A. Martinez-Agosto *et al.*, "Clinical application of next-generation sequencing to the practice of neurology," pp. 492–503, 5 2019.

[6] "Next-generation sequencing (ngs) | explore the technology." [Online]. Available: https://www.illumina.com/science/technology/next-generation-sequencing.html

[7] "Welcome to ion torrent next-generation sequencing - es." [Online]. Available: //www.thermofisher.com/es/es/home/life-science/sequencing/next-generation-sequencing/ion-torrent-next-generation-sequencing-workflow/ion-torrent-next-generation-sequencing-run-sequence/

ion-s5-ngs-targeted-sequencing/welcome-ion-torrent-next-generation-sequencing.html

[8] "Ion reporter | thermo fisher scientific." [Online]. Available: https://ionreporter.thermofisher.com/ir/

[9] "Ion reporter software - es." [Online]. Available: //www.thermofisher.com/es/es/home/life-science/sequencing/next-generation-sequencing/ion-torrent-next-generation-sequencing-workflow/ion-torrent-next-generation-sequencing-data-analysis-workflow/ion-reporter-software.html

[10] "Illumina dragen bio-it platform| variant calling secondary genomic analysis software tool." [Online]. Available: https://www.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html

[11] "Informatics products." [Online]. Available: https://www.illumina.com/products/by-type/informatics-products.html

[12] "Roche - roche 454 life sciences and softgenetics sign co-promotion agreement for next-gen sequencing software tools." [Online]. Available: https://www.roche.com/de/media/releases/med-cor-2012-05-09t.htm

[13] "Softgenetics - software powertools for genetic analysis." [Online]. Available: https://softgenetics.com/NextGENe.php

[14] "Brb-seqtools." [Online]. Available: https://brb.nci.nih.gov/seqtools/

[15] "Treva: Vm for targeted/exome sequencing - peter maccallum cancer centre." [Online]. Available: http://bioinformatics.petermac.org/treva/

[16] "Diagram maker | software de diagramas en línea | creately." [Online]. Available: https://creately.com/es/home/

[17] "Types of variants | garvan institute of medical research." [Online]. Available: https://www.garvan.org.au/research/kinghorn-centre-for-clinical-genomics/learn-about-genomics/for-gp/genetics-refresher-1/types-of-variants

[18] "Fusion gene gene expression chimeric gene cancer png, clipart, angle, area, blue, cancer, cancer cell free png download." [Online]. Available: https://imgbin.com/png/CFGpA2jy/fusion-gene-gene-expression-chimeric-gene-cancer-png

[19] "Cutadapt — cutadapt 3.4 documentation." [Online]. Available: https://cutadapt.readthedocs.io/en/stable/

[20] M. Martin, "Cutadapt removes adapter sequences from high-throughput sequencing reads," *EMBnet.journal*, vol. 17, p. 10, 5 2011.

[21] "Thermo fisher scientific - es." [Online]. Available: https://www.thermofisher.com/es/es/home.html

[22] "Prinseq @ sourceforge.net." [Online]. Available: http://prinseq.sourceforge.net/

[23] "Github - spabinger/prinseq_parallel: Parallel / multithreading / multiple threads version of prinseq." [Online]. Available: https://github.com/spabinger/prinseq_parallel

[24] "Babraham bioinformatics - fastqc a quality control tool for high throughput sequence data." [Online]. Available: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

[25] "Multiqc." [Online]. Available: https://multiqc.info/

[26] P. Ewels, M. Magnusson, S. Lundin, and M. Käller, "Multiqc: Summarize analysis results for multiple tools and samples in a single report," *Bioinformatics*, vol. 32, pp. 3047–3048, 10 2016.

[27] "Index of /goldenpath/hg19/bigzips." [Online]. Available: https://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/

[28] H. Lee, K. W. Lee, T. Lee *et al.*, "Performance evaluation method for read mapping tool in clinical panel sequencing," *Genes and Genomics*, vol. 40, pp. 189–197, 2 2018. [Online]. Available: https://link.springer.com/article/10.1007/s13258-017-0621-9

[29] S. Thankaswamy-Kosalai, P. Sen, and I. Nookaew, "Evaluation and assessment of read-mapping by multiple next-generation sequencing aligners based on genome-wide characteristics," *Genomics*, vol. 109, pp. 186–191, 7 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28286147/

[30] "Ngs read mapper comparison." [Online]. Available: https://www.ecseq.com/support/benchmark.html

[31] A. Hatem, D. Bozdağ, A. E. Toland, and Ümit V. Çatalyürek, "Benchmarking short sequence mapping tools," *BMC Bioinformatics*, vol. 14, p. 184, 6 2013. [Online]. Available: http://www.biomedcentral.com/1471-2105/14/184

[32] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows-wheeler transform," *Bioinformatics*, vol. 25, pp. 1754–1760, 7 2009.

[33] "Burrows-wheeler aligner." [Online]. Available: http://bio-bwa.sourceforge.net/

[34] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with bowtie 2," *Nature Methods*, vol. 9, pp. 357–359, 4 2012. [Online]. Available: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml.

[35] B. Langmead, C. Wilks, V. Antonescu, and R. Charles, "Scaling read aligners to hundreds of threads on general-purpose processors," *Bioinformatics*, vol. 35, pp. 421–432, 2 2019. [Online]. Available: http://bowtie-bio.sourceforge.net/bowtie2.HISAT:http://www.ccb.jhu.edu/software/hisat

[36] "Bowtie 2: fast and sensitive read alignment." [Online]. Available: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

[37] T. D. Wu, J. Reeder, M. Lawrence *et al.*, "Gmap and gsnap for genomic sequence alignment: Enhancements to speed, accuracy, and functionality," pp. 283–334, 2016. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27008021/

[38] Y. Liao, G. K. Smyth, and W. Shi, "The subread aligner: Fast, accurate and scalable read mapping by seed-and-vote," *Nucleic Acids Research*, vol. 41, p. e108, 5 2013. [Online]. Available: /pmc/articles/PMC3664803//pmc/articles/PMC3664803/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC3664803/

[39] "The subread package." [Online]. Available: http://subread.sourceforge.net/

[40] D. Kim, J. M. Paggi, C. Park, C. Bennett, and S. L. Salzberg, "Graph-based genome alignment and genotyping with hisat2 and hisat-genotype," *Nature Biotechnology*, vol. 37, pp. 907–915, 8 2019.

[41] D. Kim, B. Langmead, and S. L. Salzberg, "Hisat: A fast spliced aligner with low memory requirements," *Nature Methods*, vol. 12, pp. 357–360, 3 2015.

[42] "Novoalign | novocraft." [Online]. Available: http://www.novocraft.com/products/novoalign/

[43] S. Marco-Sola, M. Sammeth, R. Guigó, and P. Ribeca, "The gem mapper: Fast, accurate and versatile alignment by filtration," *Nature Methods*, vol. 9, pp. 1185–1188, 12 2012. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/23103880/

[44] H. N. Lin and W. L. Hsu, "Kart: A divide-and-conquer algorithm for ngs read alignment," *Bioinformatics*, vol. 33, pp. 2281–2287, 8 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28379292/

[45] "Samtools." [Online]. Available: http://www.htslib.org/

[46] "Baserecalibrator – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360036898312-BaseRecalibrator

[47] "Base quality score recalibration (bqsr) – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360035890531-Base-Quality-Score-Recalibration-BQSR-

[48] "Snp - ncbi." [Online]. Available: https://www.ncbi.nlm.nih.gov/snp/?cmd=search

[49] S. T. Sherry, M. H. Ward, M. Kholodov *et al.*, "Dbsnp: The ncbi database of genetic variation," *Nucleic Acids Research*, vol. 29, pp. 308–311, 1 2001. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/11125122/

[50] "gnomad." [Online]. Available: https://gnomad.broadinstitute.org/

[51] K. J. Karczewski, L. C. Francioli, G. Tiao, B. B. Cummings *et al.*, "The mutational constraint spectrum quantified from variation in 141,456 humans," *Nature*, vol. 581, pp. 434–443, 5 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2308-7

[52] F. García-Alcalde, K. Okonechnikov, J. Carbonell *et al.*, "Qualimap: Evaluating next-generation sequencing alignment data," *Bioinformatics*, vol. 28, pp. 2678–2679, 10 2012.

[53] J. A. Molina-Mora and M. Solano-Vargas, "Set-theory based benchmarking of three different variant callers for targeted sequencing," *BMC Bioinformatics*, vol. 22, 12 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33413082/

[54] Q. Wang, V. Kotoula, P. C. Hsu *et al.*, "Comparison of somatic variant detection algorithms using ion torrent targeted deep sequencing data," *BMC Medical Genomics*, vol. 12, pp. 1–11, 12 2019. [Online]. Available: https://doi.org/10.1186/s12920-019-0636-y

[55] X. Bian, B. Zhu, M. Wang *et al.*, "Comparing the performance of selected variant callers using synthetic data and genome segmentation," *BMC Bioinformatics*, vol. 19, pp. 1–11, 11 2018. [Online]. Available: https://doi.org/10.1186/s12859-018-2440-7

[56] Z. K. Liu, Y. K. Shang, Z. N. Chen, and H. Bian, "A three-caller pipeline for variant analysis of cancer whole-exome sequencing data," *Molecular Medicine Reports*, vol. 15, pp. 2489–2494, 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28447726/

[57] R. Bohnert, S. Vivas, and G. Jansen, "Comprehensive benchmarking of snv callers for highly admixed tumor data," *PLoS ONE*, vol. 12, 10 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29020110/

[58] "Mutect2 – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360037593851-Mutect2

[59] "Haplotypecaller – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360037225632-HaplotypeCaller

[60] D. C. Koboldt, Q. Zhang, D. E. Larson *et al.*, "Varscan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing," *Genome Research*, vol. 22, pp. 568–576, 3 2012.

[61] D. C. Koboldt, K. Chen, T. Wylie *et al.*, "Varscan: Variant detection in massively parallel sequencing of individual and pooled samples," *Bioinformatics*, vol. 25, pp. 2283–2285, 2009.

[62] "Varscan - variant detection in massively parallel sequencing data." [Online]. Available: http://varscan.sourceforge.net/

[63] Z. Lai, A. Markovets, M. Ahdesmaki *et al.*, "Vardict: A novel and versatile variant caller for next-generation sequencing in cancer research," *Nucleic Acids Research*, vol. 44, 6 2016. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27060149/

[64] "Github - astrazeneca-ngs/vardictjava: Vardict java port." [Online]. Available: https://github.com/AstraZeneca-NGS/VarDictJava

[65] E. Garrison and G. Marth, "Haplotype-based variant detection from short-read sequencing," 7 2012. [Online]. Available: http://arxiv.org/abs/1207.3907

[66] "Variant calling with freebayes | in-depth-ngs-data-analysis-course." [Online]. Available: https://hbctraining.github.io/In-depth-NGS-Data-Analysis-Course/sessionVI/lessons/02_variant-calling.html

[67] "Freebayes variant calling workflow for dna-seq - bioinformatics workbook." [Online]. Available: https://bioinformaticsworkbook.org/dataAnalysis/VariantCalling/freebayes-dnaseq-workflow.html#gsc.tab=0

[68] "Lofreq · fast and sensitive variant calling from next-gen sequencing data." [Online]. Available: https://csb5.github.io/lofreq/

[69] M. Callari, S.-J. Sammut, L. D. Mattos-Arruda *et al.*, "Intersect-then-combine approach: improving the performance of somatic variant calling in whole exome sequencing data using multiple aligners and callers," *Genome Medicine*, vol. 9, p. 35, 12 2017. [Online]. Available: http://genomemedicine.biomedcentral.com/articles/10.1186/s13073-017-0425-1

[70] "bcftools." [Online]. Available: http://samtools.github.io/bcftools/bcftools.html#norm

[71] "Vcftools: Perl tools and api." [Online]. Available: http://vcftools.sourceforge.net/perl_module.html#vcf-isec

[72] "bcftools." [Online]. Available: http://samtools.github.io/bcftools/bcftools.html#concat

[73] "Usage — vcf toolz 1.2.0 documentation." [Online]. Available: https://vcftoolz.readthedocs.io/en/latest/usage.html#compare

[74] K. Wang, M. Li, and H. Hakonarson, "Annovar: Functional annotation of genetic variants from high-throughput sequencing data," *Nucleic Acids Research*, vol. 38, pp. e164–e164, 7 2010. [Online]. Available: https://academic.oup.com/nar/article/38/16/e164/1749458

[75] "Annovar documentation." [Online]. Available: https://annovar.openbioinformatics.org/en/latest/

[76] K. Pagel, R. Kim, K. Moad *et al.*, "Opencravat, an open source collaborative platform for the annotation of human genetic variation," *bioRxiv*, p. 794297, 10 2019. [Online]. Available: https://www.biorxiv.org/content/10.1101/794297v1https://www.biorxiv.org/content/10.1101/794297v1.abstract

[77] "Open cravat." [Online]. Available: https://opencravat.org/

[78] "Myvariant.info." [Online]. Available: https://myvariant.info/

[79] J. Xin, A. Mark, C. Afrasiabi *et al.*, "High-performance web services for querying gene and variant annotation," *Genome Biology*, vol. 17, pp. 1–7, 5 2016. [Online]. Available: https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0953-9

[80] "Bioconductor - myvariant." [Online]. Available: https://bioconductor.org/packages/release/bioc/html/myvariant.html

[81] "Filter-based annotation - annovar documentation." [Online]. Available: https://annovar.openbioinformatics.org/en/latest/user-guide/filter/#exac-annotations

[82] K. J. Karczewski, B. Weisburd, B. Thomas *et al.*, "The exac browser: Displaying reference data information from over 60 000 exomes," *Nucleic Acids Research*, vol. 45,

pp. D840–D845, 1 2017. [Online]. Available: https://academic.oup.com/nar/article/45/D1/D840/2572071

[83] "1000 genomes | a deep catalog of human genetic variation." [Online]. Available: https://www.internationalgenome.org/

[84] G. Glusman, J. Caballero, D. E. Mauldin *et al.*, "Kaviar: An accessible system for testing snv novelty," pp. 3216–3217, 11 2011.

[85] "Kaviar genomic variant database | snp database | snv database | isb." [Online]. Available: http://db.systemsbiology.net/kaviar/

[86] "Sift - predict effects of nonsynonmous / missense variants." [Online]. Available: https://sift.bii.a-star.edu.sg/

[87] "Polyphen-2: prediction of functional effects of human nssnps." [Online]. Available: http://genetics.bwh.harvard.edu/pph2/

[88] "Mutationtaster." [Online]. Available: http://www.mutationtaster.org/

[89] H. Carter, S. Chen, L. Isik *et al.*, "Cancer-specific high-throughput annotation of somatic mutations: Computational prediction of driver missense mutations," *Cancer Research*, vol. 69, pp. 6660–6667, 8 2009. [Online]. Available: /pmc/articles/PMC2763410//pmc/articles/PMC2763410/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC2763410/

[90] H. Carter, C. Douville, P. D. Stenson *et al.*, "Identifying mendelian disease genes with the variant effect scoring tool." *BMC genomics*, vol. 14 Suppl 3, p. S3, 2013. [Online]. Available: /pmc/articles/PMC3665549//pmc/articles/PMC3665549/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC3665549/

[91] "Cosmic | catalogue of somatic mutations in cancer." [Online]. Available: https://cancer.sanger.ac.uk/cosmic

[92] S. A. Forbes, D. Beare, H. Boutselakis *et al.*, "Cosmic: Somatic cancer genetics at high-resolution," *Nucleic Acids Research*, vol. 45, pp. D777–D783, 1 2017.

[93] "Clinvar." [Online]. Available: https://www.ncbi.nlm.nih.gov/clinvar/

[94] "Civic: Home." [Online]. Available: https://civicdb.org/home

[95] "Oncokb." [Online]. Available: https://www.oncokb.org/

[96] D. Chakravarty, J. Gao, S. Phillips *et al.*, "Oncokb: A precision oncology knowledge base," *JCO Precision Oncology*, pp. 1–16, 11 2017.

[97] "Varsome the human genomics community." [Online]. Available: https://varsome.com/

[98] J. M. Moreno-Cabrera, J. del Valle, E. Castellanos *et al.*, "Evaluation of cnv detection tools for ngs panel data in genetic diagnostics," *European Journal of Human Genetics*, vol. 28, pp. 1645–1655, 12 2020. [Online]. Available: https://doi.org/10.1038/s41431-020-0675-z

[99] L. Zhao, H. Liu, X. Yuan *et al.*, "Comparative study of whole exome sequencing-based copy number variation detection tools," *BMC Bioinformatics*, vol. 21, pp. 1–10, 3 2020. [Online]. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-3421-1

[100] I. Roca, L. González-Castro, H. Fernández *et al.*, "Free-access copy-number variant detection tools for targeted next-generation sequencing data," pp. 114–125, 1 2019. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/31097148/

[101] "Cnvkit: Genome-wide copy number from high-throughput sequencing — cnvkit 0.9.8 documentation." [Online]. Available: https://cnvkit.readthedocs.io/en/stable/

[102] E. Talevich, A. H. Shain, T. Botton, and B. C. Bastian, "Cnvkit: Genome-wide copy number detection and visualization from targeted dna sequencing," *PLoS Computational Biology*, vol. 12, 4 2016.

[103] "Convading user guide." [Online]. Available: http://molgenis.github.io/CoNVaDING/

[104] L. F. Johansson, F. van Dijk, E. N. de Boer *et al.*, "Convading: Single exon variation detection in targeted ngs data," *Human Mutation*, vol. 37, pp. 457–464, 5 2016. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/26864275/

[105] "Exomedepth: Calls copy number variants from targeted sequence data version 1.1.15 from cran." [Online]. Available: https://rdrr.io/cran/ExomeDepth/

[106] "Bioconductor - panelcn.mops." [Online]. Available: https://www.bioconductor.org/packages/release/bioc/html/panelcn.mops.html

[107] G. Povysil, A. Tzika, J. Vogt *et al.*, "panelcn.mops: Copy-number detection in targeted ngs panel data for clinical diagnostics," *Human Mutation*, vol. 38, pp. 889–897, 7 2017.

[108] "Control-freec: Copy number and allelic content caller." [Online]. Available: http://boevalab.inf.ethz.ch/FREEC/

[109] V. Boeva, T. Popova, K. Bleakley *et al.*, "Control-freec: A tool for assessing copy number and allelic content using next-generation sequencing data," *Bioinformatics*, vol. 28, pp. 423–425, 2 2012.

[110] B. J. Haas, A. Dobin, B. Li *et al.*, "Accuracy assessment of fusion transcript detection via read-mapping and de novo fusion transcript assembly-based methods," *Genome Biology*, vol. 20, 10 2019.

[111] S. Kumar, A. D. Vo, F. Qin, and H. Li, "Comparative assessment of methods for the fusion transcripts detection from rna-seq data," *Scientific Reports*, vol. 6, 2 2016.

[112] S. Uhrig, J. Ellermann, T. Walther *et al.*, "Accurate and efficient detection of gene fusions from rna sequencing data," *Genome Research*, vol. 31, pp. 448–460, 1 2021.

[113] "Home - arriba." [Online]. Available: https://arriba.readthedocs.io/en/latest/

[114] D. Nicorici, M. Satalan, H. Edgren *et al.*, "Fusioncatcher - a tool for finding somatic fusion genes in paired-end rna-sequencing data," *bioRxiv*, p. 011650, 11 2014. [Online]. Available: http://code.google.com/p/fusioncatcher/.

[115] B. Haas, A. Dobin, N. Stransky *et al.*, "Star-fusion: Fast and accurate fusion transcript detection from rna-seq," *bioRxiv*, p. 120295, 3 2017. [Online]. Available: https://doi.org/10.1101/120295

[116] K. Cibulskis, M. S. Lawrence, S. L. Carter *et al.*, "Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples," *Nature Biotechnology*, vol. 31, pp. 213–219, 3 2013. [Online]. Available: http://www.

[117] "Somatic short variant discovery (snvs + indels) – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360035894731-Somatic-short-variant-discovery-SNVs-Indels-

[118] "Learnreadorientationmodel – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360051305331-LearnReadOrientationModel

[119] "Getpileupsummaries – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360037593451-GetPileupSummaries

[120] "Calculatecontamination – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360036888972-CalculateContamination

[121] "Filtermutectcalls – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360036856831-FilterMutectCalls

[122] "Germline short variant discovery (snps + indels) – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360035535932-Germline-short-variant-discovery-SNPs-Indels-

[123] "Genotypegvcfs – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360037057852-GenotypeGVCFs

[124] "Developing low frequency filters for cancer variant calling using vardict – blue collar bioinformatics." [Online]. Available: http://bcb.io/2016/04/04/vardict-filtering/

[125] J. T. D. Dunnen, "Describing sequence variants using hgvs nomenclature," pp. 243–251, 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27822869/

[126] "Whole-genome sequencing and targeted amplicon capture — cnvkit 0.9.8 documentation." [Online]. Available: https://cnvkit.readthedocs.io/en/stable/nonhybrid.html#targeted-amplicon-sequencing-tas

[127] "Home ncip/trinity_ctat wiki github." [Online]. Available: https://github.com/NCIP/Trinity_CTAT/wiki

[128] A. Frankish, M. Diekhans, A. M. Ferreira *et al.*, "Gencode reference annotation for the human and mouse genomes," *Nucleic Acids Research*, vol. 47, pp. D766–D773, 1 2019. [Online]. Available: https://academic.oup.com/nar/article/47/D1/D766/5144133

[129] "Smc-rna - syn2813589 - wiki." [Online]. Available: https://www.synapse.org/#!Synapse:syn2813589/wiki/401435

[130] K. Eilbeck, S. E. Lewis, C. J. Mungall *et al.*, "The sequence ontology: a tool for the unification of genome annotations." *Genome biology*, vol. 6, pp. 1–12, 4 2005. [Online]. Available: http://genomebiology.com/2005/6/5/R44

[131] "Haplotypecallerspark (beta) – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360037433931-HaplotypeCallerSpark-BETA-

[132] L. Song, W. Huang, J. Kang, Y. Huang, H. Ren, and K. Ding, "Comparison of error correction algorithms for ion torrent pgm data: Application to hepatitis b virus," *Scientific Reports*, vol. 7, pp. 1–11, 12 2017. [Online]. Available: www.nature.com/scientificreports

[133] A. Mohsen, J. Park, Y. A. Chen, H. Kawashima, and K. Mizuguchi, "Impact of quality trimming on the efficiency of reads joining and diversity analysis of illumina paired-end reads in the context of qiime1 and qiime2 microbiome analysis frameworks," *BMC Bioinformatics*, vol. 20, pp. 1–10, 11 2019. [Online]. Available: https://doi.org/10.1186/s12859-019-3187-5

[134] "Termofisher manual." [Online]. Available: https://tools.thermofisher.com/content/sfs/manuals/CO25176_0512.pdf

[135] K. Day, J. Song, and D. Absher, "Targeted sequencing of large genomic regions with catch-seq," *PLoS ONE*, vol. 9, 10 2014.

[136] H. M. Schilbert, A. Rempel, and B. Pucker, "Comparison of read mapping and variant calling tools for the analysis of plant ngs data," *Plants*, vol. 9, p. 439, 4 2020. [Online]. Available: www.mdpi.com/journal/plants

[137] B. N. Keel and W. M. Snelling, "Comparison of burrows-wheeler transform-based mapping algorithms used in high-throughput whole-genome sequencing: Application to illumina data for livestock genomes 1," *Frontiers in Genetics*, vol. 9, p. 35, 2 2018. [Online]. Available: www.frontiersin.org

[138] J. O'Rawe, T. Jiang, G. Sun *et al.*, "Low concordance of multiple variant-calling pipelines: Practical implications for exome and genome sequencing," *Genome Medicine*, vol. 5, pp. 1–18, 3 2013. [Online]. Available: http://genomemedicine.com/content/5/3/28

[139] "Cobertura - leer métricas de profundidad - gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360035532112-Coverage-Read-depth-metrics

[140] "Allele depth (ad) is lower than expected – gatk." [Online]. Available: https://gatk.broadinstitute.org/hc/en-us/articles/360035532252-Allele-Depth-AD-is-lower-than-expected

[141] "Gatk 4 mutect2 gives too much low depth variants." [Online]. Available: https://www.biostars.org/p/354369/

[142] "Sift help." [Online]. Available: https://sift.bii.a-star.edu.sg/www/SIFT_help.html

[143] "docs [polyphen-2 wiki]." [Online]. Available: http://genetics.bwh.harvard.edu/pph2/dokuwiki/docs

[144] C. Dong, P. Wei, X. Jian *et al.*, "Comparison and integration of deleteriousness prediction methods for nonsynonymous snvs in whole exome sequencing studies," *Human Molecular Genetics*, vol. 24, pp. 2125–2137, 4 2015. [Online]. Available: /pmc/articles/PMC4375422//pmc/articles/PMC4375422/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC4375422/

[145] "Pathogenicity predictions." [Online]. Available: https://www.ensembl.org/info/genome/variation/prediction/protein_function.html

[146] "Mutationtaster - faqs." [Online]. Available: http://www.mutationtaster.org/info/FAQs.html

[147] "Descripción general de chasm - chasm software wiki." [Online]. Available: http://wiki.chasmsoftware.org/index.php?title=CHASM_Overview&oldid=816

[148] S. Schbath, V. Martin, M. Zytnicki *et al.*, "Mapping reads on a genomic sequence: An algorithmic overview and a practical comparative analysis," *Journal of Computational Biology*, vol. 19, pp. 796–813, 6 2012. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/22506536/

Apéndice A
Manual de uso

Para poder ejecutar el flujo de trabajo construido en este proyecto, el primer y fundamental requisito es disponer de los scripts **pipeline_picasso_ampliacion.sh** y **configure.sh**, así como las carpetas asociadas a los archivos de referencia y las herramientas instaladas localmente. Una vez logrado esto, el usuario deberá seguir los tres pasos que se comentan a continuación.

A.1. Modificación del archivo de configuración

El archivo de configuración **configure.sh** se encuentra diseñado para ser modificado por usuarios sin necesidad de que posean conocimientos informáticos. Para ello, el archivo comienza con una sección denominada "reglas" destinada a proporcionar información básica acerca de cómo editar correctamente el valor de las variables. Consta de 5 normas fundamentales:

1. Todas las líneas que comienzan con una almohadilla son consideradas comentarios cuyo contenido no afecta a la ejecución del programa.
2. Es muy importante no dejar espacios entre la variable, el igual y su valor:
    - Bien escrito: variable=valor
    - Mal escrito: variable = valor, variable= valor, variable =valor
3. Los valores distintos a true y false van siempre entre comillas y con todas las letras en minúsculas.
4. A los valores de tipo vector nunca se les quita los paréntesis, incluso cuando sólo se especifique un elemento.

5. No tocar ninguna variable que se encuentre dentro de una sentencia tipo if.

Una vez que el usuario comprende el funcionamiento del script, ya se encuentra preparado para manipular la sección de parámetros. Con el fin de facilitar la edición de variables, se ha elaborado un diseño formado por un total de 13 cuestiones para las que se proporciona en todo momento las posibles respuestas contempladas.

1. ¿Quieres generar los informes de calidad de las lecturas iniciales?

```
# OPCIONES: true false
bool_fastq_reports_iniciales=true
```

2. ¿Quieres generar los informes de calidad de las lecturas ya procesadas?

```
# OPCIONES: true false
bool_fastq_reports_procesadas=true
```

3. ¿Quieres generar los informes comparativos del antes y el después de las lecturas? En caso de que sí, las dos variables anteriores se pondrán a true automáticamente.

```
# OPCIONES: true false
bool_multiqc=true

if [ $bool_multiqc == true ]
then
  bool_fastq_reports_iniciales=true
  bool_fastq_reports_procesadas=true
fi
```

4. ¿Qué alineadores quieres emplear?

    - Si se elige sólo 1, no se realizará la limpieza de variantes mediante la comparación de resultados de distintos mapeadores.
    - Si se eligen 2, las variantes de cada variant caller surgirán de la intersección estricta entre ambos mapeadores.

- Si se eligen 3 o más, las variantes de cada variant caller surgirán de la intersección (dejando como margen la abstención en algún fichero) entre todos los mapeadores.

```
# OPCIONES: "bwa" "bowtie2" "gmap" "subread" "hisat2" "novoalign" "gem3" "
    kart"
alineadores_dna=("bwa" "hisat2" "gem3")
```

5. ¿Quieres generar los informes de calidad de los distintos mapeadores empleados?

   - Si se ha elegido sólo un mapeador, se elaborará un único informe con sus correspondientes estadísticas.
   - Si se han elegido 2 o más mapeadores, se elaborará un informe individual para cada uno y uno conjunto que los compare.

```
# OPCIONES: true false
bool_bam_reports=true
```

6. ¿Qué variant callers quieres utilizar? Los resultados de cada variant caller (intersección de mapeadores) se unirán en un mismo vcf para proceder a su posterior anotación.

```
# OPCIONES: "gatk_mutect2" "gatk_haplotypecaller" "varscan" "vardict" "
    freebayes" "lofreq"
variant_callers=("gatk_mutect2" "gatk_haplotypecaller" "varscan" "vardict"
    "freebayes" "lofreq")
```

7. ¿Quieres realizar diagramas de Venn comparando los resultados de los mapeadores para cada variant caller? Sólo podrán realizarse si el número de alineadores es mayor que 1 y menor que 7.

```
# OPCIONES: true false
bool_venn_diagrams_mappers=true

if [ ${#alineadores_dna[@]} = 1 ] || [ ${#alineadores_dna[@]}  gt  6 ]
then
  bool_venn_diagrams_mappers=false
fi
```

8. ¿Quieres realizar diagramas de Venn comparando las intersecciones de distintos variant callers? Sólo podrán realizarse si el número de variant callers es mayor que 1 y menor que 7.

```
# OPCIONES: true false
bool_venn_diagrams_vcs=true

if [ ${#variant_callers[@]} = 1 ] || [ ${#variant_callers[@]} gt 6 ]
then
  bool_venn_diagrams_vcs=false
fi
```

9. ¿Quieres ejecutar el flujo asociado a la detección de CNVs? Si la respuesta es false, ignora las preguntas 10 y 11.

```
# OPCIONES: true false
cnv=true
```

10. ¿Qué alineador quieres escoger como preferente para ser utilizado en la detección de CNVs? Para ello, el alineador tiene que estar incluido dentro del vector alineadores_dna, de lo contrario, no se ejecutará el script.

```
# OPCIONES: "bwa" "bowtie2" "gmap" "subread" "hisat2" "novoalign" "gem3" "
    kart"
alineador_dna_preferente="bwa"

if [ $cnv == true ]
then
  [[ $alineadores_dna =~ (|[[:space:]])$alineador_dna_preferente($|[[:space
    :]]) ]] || exit
fi
```

11. ¿Qué herramientas quieres emplear para la detección de CNVs?

```
# Si el número de muestras es inferior a 3, OPCIONES: "cnvkit" "
    controlfreec"
# Si el número de muestras es igual o mayor que 3, OPCIONES: "cnvkit" "
    convading" "exomedepth" "panelcnmops"
cnv_detectors=("cnvkit" "convading" "exomedepth" "panelcnmops")
```

12. ¿Quieres ejecutar el flujo asociado a la detección de Fusiones Génicas? Si la respuesta es false, ignora las pregunta 13.

```
# OPCIONES: true false
gene_fusion=true
```

13. ¿Qué herramientas quieres emplear para la detección de Fusiones Génicas?

```
# OPCIONES: "fusioncatcher" "arriba" "star_fusion"
gene_fusion_detectors=("arriba" "star_fusion" "fusioncatcher")
```

A.2. Posicionamiento y renombrado de los archivos FASTQ

Tras personalizar la ejecución del programa a través del fichero de configuración, el siguiente paso consiste en proporcionar correctamente los datos de entrada. Esto significa que el usuario debe trasladar los archivos FASTQ de las muestras situadas en su almacenamiento local al directorio de Picasso. Para ello, se puede hacer uso de **sftp** o aplicaciones con interfaz gráfica como **FileZilla**.

Para que la ejecución sea correcta, estos ficheros deben situarse en el mismo directorio en el que se encuentran las carpetas de los archivos de referencia y las herramientas instaladas localmente. Además, tal y como se menciona en la memoria, los nombres de estos FASTQ deben seguir un patrón específico. Todos ellos deben comenzar con el identificador del paciente al que pertenecen seguido de un guión bajo con el tipo de muestra correspondiente (dna o rna).

A.3. Ejecución del script

Por último, el usuario debe acceder a la terminal para poner en marcha el flujo de trabajo construido. Para ello, iniciará sesión en Picasso, accederá a la nueva ampliación de la supercomputadora y ejecutará el script principal de este proyecto:

```
ssh user_name@picasso.scbi.uma.es
ssh sd001
sbatch pipeline_picasso_ampliacion.sh
```

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA